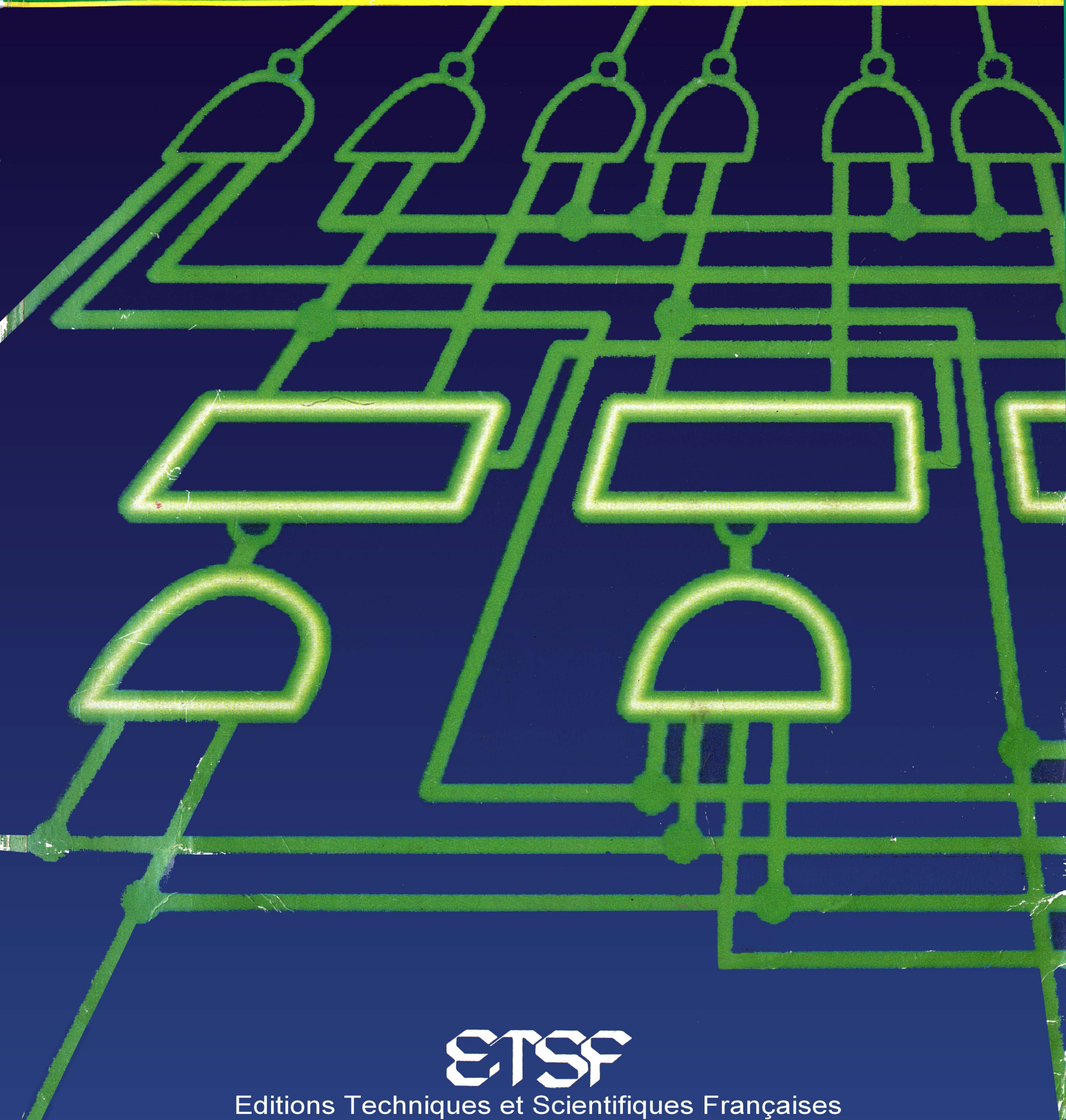
C. PANETTO

GUIDE PRATIQUE DES SYSTEMES LOGIQUES



ETSF
Editions Techniques et Scientifiques Françaises

Christian PANETTO

Agrégé de génie électrique
Professeur au
Lycée Technique Vilgénis à Massy
et au
Centre de Formation continue

GUIDE PRATIQUE DES SYSTÈMES LOGIQUES

Techniques numériques ;
Circuits intégrés logiques ;
Fonctions combinatoires et séquentielles ;
Mémoires ;
Adaptation des signaux ;
Convertisseurs A/N et N/A ;
Visualisation.



Editions Techniques
et Scientifiques Françaises
2 à 12, rue de Bellevue 75940 Paris cedex 19
Tél. (1) 42.00.33.05

Imprimé en France par Berger-Levrault, Nancy

Dépôt légal : juillet 1986
Éditeur n° 487 - Imprimeur : 778622
I.S.B.N. 2 85535 159 6

Introduction

Le développement actuel de la micro-informatique, des automatismes, de la robotique s'appuie sur une infrastructure technologique qui le rend possible.

La connaissance des circuits de logique et des circuits d'interface est nécessaire pour qui veut aborder ces domaines sur le plan matériel. Cet ouvrage, par son approche progressive et didactique aborde les deux aspects fondamentaux de ces types de circuits : la fonction assurée et l'électronique qui réalise la fonction.

Après quelques généralités fondamentales, l'auteur aborde les systèmes de numération qui permettent le dialogue entre l'homme et la machine.

Ensuite l'étude des systèmes combinatoires et séquentiels par des exemples permet de lever le voile sur les fonctions couramment utilisées dans les systèmes logiques ; telles que multiplexage, décodage ou addition par exemple pour les uns, mémorisation, décalage, comptage et décomptage pour les autres.

L'approche technologique des différentes familles de mémoire nous permet, pour la bonne compréhension de leur fonctionnement, de choisir correctement le type de circuit en fonction du problème à traiter ou de réaliser des extensions mémoires pour des systèmes informatiques auxquels il est nécessaire d'augmenter la capacité « mémoire », par exemple.

Il est aussi indispensable de savoir « adapter » les signaux électriques pour transmettre une information d'un type de circuit à un autre. C'est le rôle des circuits d'interface qui sont abordés dans un chapitre particulier où les principes essentiels et les circuits qui s'y rattachent sont vus.

La conversion numérique-analogique et analogique-numérique est nécessaire pour qui veut traiter numériquement des grandeurs analogiques. Après une étude théorique des différents types de conversion, des exemples de circuits intégrés sont donnés.

Les sytèmes de visualisation sont abordés progressivement ainsi que leurs organes de commande. C'est par cette «interface» machine-homme que l'auteur termine l'analyse de « l'électronique des systèmes logiques » puis fait une synthèse partielle en traitant un problème concret.

Tous les domaines sont abordés sous le double aspect de la logique et de l'électronique, car c'est en connaissant les deux volets, la fonction assurée et l'électronique qui la sert, que l'on peut maîtriser les systèmes logiques qui sont au cœur des technologies modernes.

I - Bases fondamentales

1. Signaux logiques

Dans un système logique, l'information traitée est de type binaire. Elle est représentée par des signaux électriques à 2 états. Ces signaux électriques diffèrent suivant la logique employée. Nous allons étudier les logiques les plus utilisées dans les systèmes combinatoires et séquentiels que nous allons définir auparavant.

2. Logique combinatoire

Les problèmes de logique combinatoire conduisent à l'établissement de « circuits combinatoires » ou de pures combinaisons dans lesquelles la notion de temps n'intervient pas. Les états des variables d'entrée sont seuls à considérer (fig. 1.1).

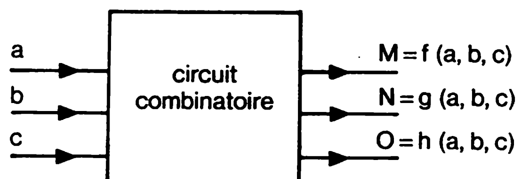


Fig. 1-1. Circuit combinatoire.

Exemple de circuit combinatoire (fig. 1.2)

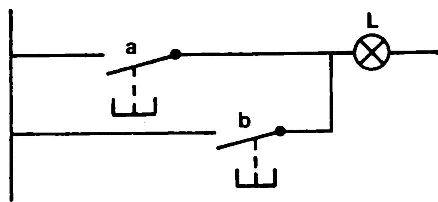


Fig. 1-2. Circuit électrique combinatoire.

L = a + b
La lampe L s'allume si on appuie sur le bouton-poussoir a ou sur le bouton-poussoir b.

3. Logique séquentielle asynchrone

Les problèmes de logique séquentielle conduisent à l'établissement de circuits dans lesquels les signaux de sortie dépendent des variables d'entrée et du temps (fig. 1-3).

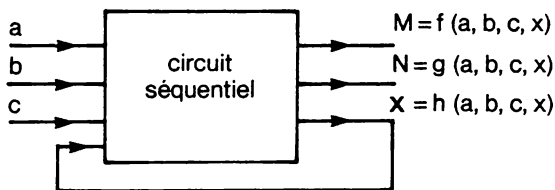


Fig. 1-3. Circuit séquentiel.

Exemple de circuit séquentiel (fig. 1.4)

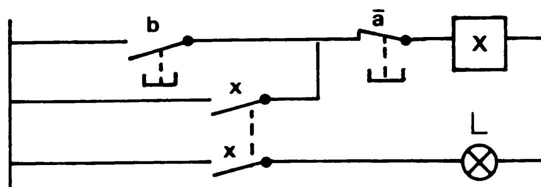


Fig. 1-4. Circuit électrique séquentiel.

La lampe s'allume si le contact x se ferme. Celui-ci ne peut se fermer que si la bobine du relais X est alimentée, c'est-à-dire si on appuie sur le bouton-poussoir b. C'est donc le bouton-poussoir b qui assurera l'allumage de la lampe L, mais après un certain temps Δt dépendant du temps de réponse du relais électromagnétique X. Le bouton-poussoir $\bar{a}$ assurera l'extinction de la lampe L. Le contact x branché en parallèle avec le bouton-poussoir b assure l'automaintien du relais X quand le bouton-poussoir b aura été relâché. On peut donc donner le schéma équivalent d'un circuit séquentiel (fig. 1-5).

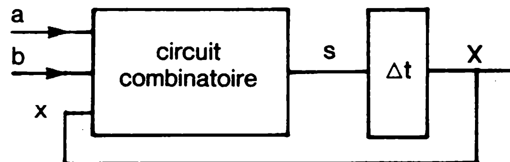


Fig. 1-5. Schéma équivalent d'un circuit séquentiel asynchrone.

S'il se présente un changement d'état d'une ou plusieurs entrées, trois cas peuvent se présenter :

1. La sortie du circuit combinatoire s ne change pas, donc X ne change pas et l'état est stable.
2. La sortie du circuit combinatoire s change d'état, donc, au bout d'un certain temps Δt, X change et nous tombons sur un état stable après être passé par un état transitoire instable.
3. Tout se passe comme précédemment mais au lieu de finir sur un état stable, le système reste perpétuellement en état instable.

Les différents événements se produisent à des instants ne dépendant que des temps d'exécution de chaque circuit du système : c'est de la **logique séquentielle asynchrone.**

4. Logique séquentielle synchrone

Le séquencement des différentes opérations logiques est **synchronisé** sur les signaux d'un générateur de rythme ou **horloge.**

De mise en œuvre plus simple, elle nécessite néanmoins l'association d'un plus grand nombre de circuits combinatoires (fig. 1-6).

Cette logique synchrone supprime les **aléas** de propagation des systèmes séquentiels asynchrones car les retards Δt dans ces derniers cir-

cuits ne sont pas identiques pour toutes les variables binaires ni pour tous les chemins conduisant d'une entrée à une sortie.

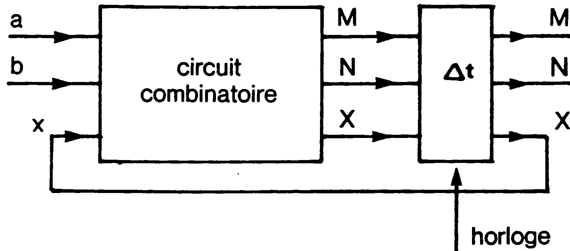


Fig. 1-6. Schéma équivalent d'un circuit séquentiel sychrone.

Dans la logique séquentielle synchrone on synchronise les évolutions des entrées secondaires.

Un rappel sur l'algèbre de Boole va nous permettre de mettre en relation les équations fondamentales booléennes et les logiques associées.

5. Algèbre de Boole

Si on considère un ensemble référentiel E à un seul élément, l'ensemble des parties de E se réduit à E et ø (ensemble vide). Sur cet ensemble à deux parties, on fait correspondre les images :

$$I(E) = 1$$
$$I(\varnothing) = 0$$

On définit ainsi un ensemble K (fig. 1-7) à deux éléments : K = (1,0).

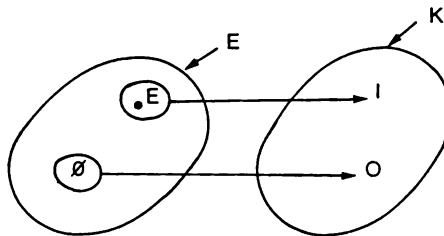


Fig. 1-7. Algèbre de Boole.

L'opération **réunion** sera notée + qu'on lit **« ou »**
L'opération **intersection** sera notée × ou · qu'on lit **« et »**.

On définit ainsi entre E et K une **application bijective.**
Sur cet ensemble K on a donc l'algèbre suivante appelée **Algèbre de Boole :**

$1+1=1$	$1 \cdot 1=1$
$1+0=1$	$1 \cdot 0=0$
$0+1=1$	$0 \cdot 1=0$
$0+0=1$	$0 \cdot 0=0$

Si $x \in K$ ($x=1$ ou $x=0$) on obtient les formules caractéristiques :

$$x+x=x \quad \text{et} \quad x \cdot x=x$$

Si on désigne par $\bar{x}$ le complémentaire de x :

$$x=1 \Longrightarrow \bar{x}=0$$

$$x=0 \Longrightarrow \bar{x}=1$$

$$\text{On a :}\ x+\bar{x}=1 \quad \text{et} \quad x \cdot \bar{x}=0$$

6. Logique à « contact »

Dans beaucoup d'applications d'électrotechnique, les circuits électriques sont réalisés suivant une logique dite « à contacts », dans laquelle on définit pour le « 0 » et le « 1 » différents états que nous allons étudier.

• Etats technologiques d'un contact électrique

Il existe deux grands types de contacts électriques :
- les contacts ouverts au repos (encore appelés « fermeture » ou « travail ») ;
- les contacts fermés au repos (encore appelés « à ouverture » ou « repos »).

Ces deux types de contacts définiront par convention deux états technologiques de contact, dont les dénominations seront :

- a si le contact est ouvert au repos : a
- $\bar{a}$ si le contact est fermé au repos : $\bar{a}$

Rappel des normes

Les contacts doivent être représentés au repos, le déplacement de l'élément mobile pouvant s'effectuer dans n'importe quel sens. Néan-

moins, sur un schéma, il est recommandé d'utiliser un même sens. La Commission électronique internationale (CEI) préconise « gauche vers droite » et « bas en haut ».

• Etats physiques d'un contact électrique

Il existe deux états physiques possibles d'un même contact électrique :

- le contact n'est pas actionné : état physique 0
- le contact est actionné : état physique 1

• Etats électriques d'un contact électrique

Ce dernier état dépend de l'état technologique et de l'état physique de ce contact. Il y a également deux états électriques possibles d'un même contact :

- le contact n'est pas passant : état électrique 0
- le contact est passant : état électrique 1

• Variables d'entrée

Ce sont tous des organes de commande (bouton-poussoir, capteur, interrupteur, etc. connus au départ et dont les changements d'état électrique permettront d'obtenir le fonctionnement désiré de l'appareil à commander en logique combinatoire.

En logique séquentielle s'ajouteront aux variables d'entrée principales les variables d'entrée secondaires, qui sont fonction des sorties secondaires du circuit.

• Logiques de sortie

La logique de sortie caractérise le fonctionnement ou le non-fonctionnement du récepteur à commander (lampe, moteur, sortie secondaire, etc.). Par convention, elle prendra :

- la valeur 0 si le récepteur est au repos ;
- la valeur 1 si le récepteur est au travail.

• Tableau d'analyse

C'est un tableau comportant n colonnes correspondant à une variable d'entrée principale. Ce tableau comporte un nombre de lignes compris entre 1 et 2^n. A chaque combinaison de variables, on fait correspondre l'état (0 ou 1) de la ou des logiques de sortie.

Exemple

Commander une lampe L (logique de sortie) de deux endroits différents à l'aide de deux interrupteurs a et b (variables d'entrée).

	a	b	L
Repos (1)	0	0	0 repos
Action sur a (2)	1	0	1 allumage
Action sur b (3)	1	1	0 extinction
Plus d'action sur a (4)	0	1	1 allumage

On a ainsi défini le cycle de fonctionnement de L.

• Analogie entre l'algèbre de Boole et les circuits électriques

Nous avons vu au paragraphe 5 que l'algèbre de Boole établissait les équations fondamentales :

$$x+\bar{x}=1 \qquad x\cdot\bar{x}=0$$

D'autre part, que l'on pouvait faire correspondre à un contact électrique ouvert au repos l'état technologique x et à un contact électrique fermé au repos l'état technologique $\bar{x}$ et que l'on caractérise le fonctionnement ou le non-fonctionnement d'un récepteur par un « 1 » ou un « 0 ».

On en déduit les schémas électriques suivants (fig. 1.8).

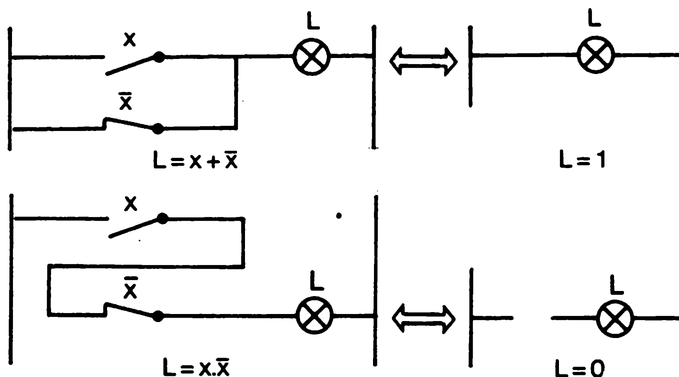


Fig. 1-8. Analogie entre l'algèbre de Boole et les circuits électriques.

Conclusion

Le « + » correspond à un branchement **parallèle.**
Le « x » correspond à un branchement **série.**

Exemple (fig. 1.9)

Schéma booléen dont l'équation logique est donnée :

$$L = \bar{a}b + c$$

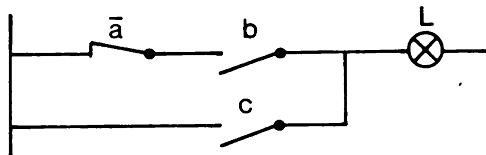


Fig. 1-9. Exemple de schéma booléen.

Equation logique dont le schéma booléen est donné (fig. 1.10)

$$L = a\,(b + \bar{c} + d)\,\bar{e} :$$

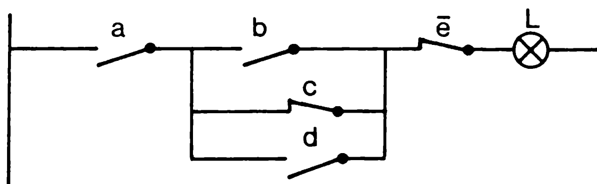


Fig. 1-10. Equation logique d'un schéma booléen.

7. Logique à niveaux

On a choisi des tensions de référence appelées « niveaux ».
Le niveau de plus haute tension positive est dit « niveau haut » noté H.

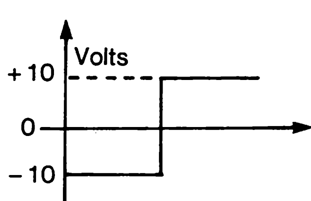


Fig. 1-11. Niveaux de tension.

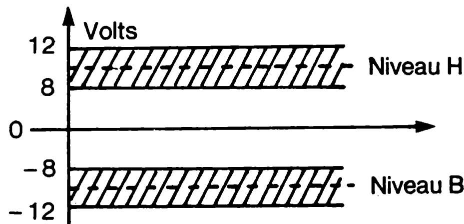


Fig. 1-12. Plage de niveaux de tension.

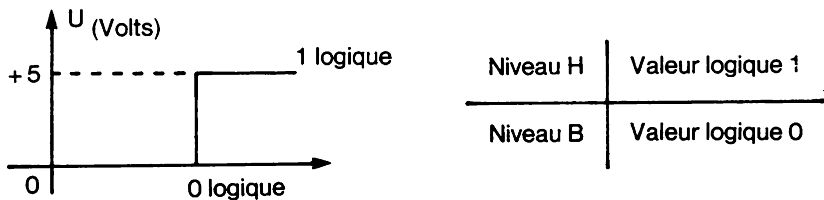
Le niveau de plus basse tension négative est dit « niveau bas » noté B (fig. 1.11).

En fait il ne s'agit pas de valeurs précises de la tension, mais de plages autour de ces valeurs (fig. 1.12).

On adopte deux conventions.

- **Logique positive :**

La tension représentant le 1 logique est supérieure à la tension représentant le 0 logique (fig. 1.13).

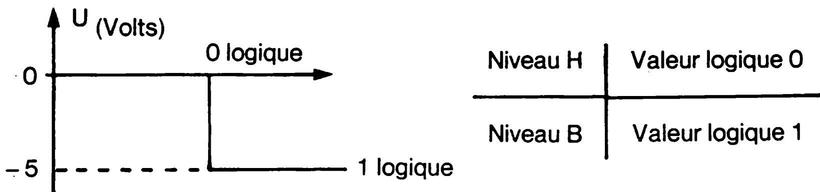


Niveau H	Valeur logique 1
Niveau B	Valeur logique 0

Fig. 1-13. Logique positive.

- **Logique négative :**

La tension représentant le 1 logique est inférieure à la tension représentant le 0 logique (fig. 1.14).



Niveau H	Valeur logique 0
Niveau B	Valeur logique 1

Fig. 1-14. Logique négative.

Remarque :

Très souvent le 0 logique est représenté par la mise à la masse soit donc une tension de 0 volt.

Suivant la technologie employée, nous serons en logique positive ou en logique négative (fig. 1.15 et 1.16).

Quand le transistor est bloqué Us = + 10 V niveau logique 1.
Quand le transistor est saturé Us = 0 V niveau logique 0.
Nous sommes en logique **positive** (fig. 1.15).

Quand le transistor est bloqué Us = – 10 V niveau logique 1.
Quand le transistor est saturé Us = 0 V niveau logique 0.
Nous sommes en logique **négative** (fig. 1.16).

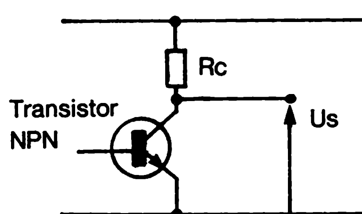


Fig. 1-15. Logique positive.

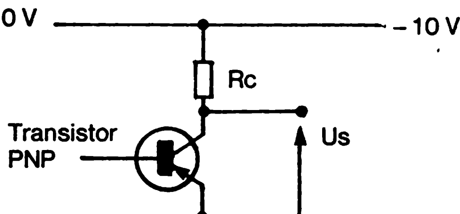


Fig. 1-16. Logique négative.

8. Logique à impulsions

Nous sommes en logique impulsionnelle lorsque la grandeur électrique représentant l'information logique dure un temps très court (fig. 1.17).

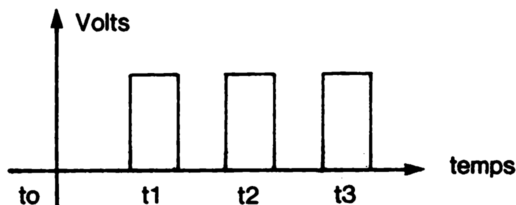


Fig. 1-17. Information logique par impulsions.

On devra lire nos informations à des intervalles de temps réguliers. Lorsque l'impulsion coïncidera avec un top de lecture, nous aurons un niveau logique 1. Lorsque le top de lecture ne coïncidera pas avec une impulsion électrique, nous aurons un niveau logique 0.

9. Transmission d'une information

- **Transmission série**

L'information codée sur n créneaux est transmise en délivrant les créneaux les uns à la suite des autres sur le même fil (fig. 1.18).

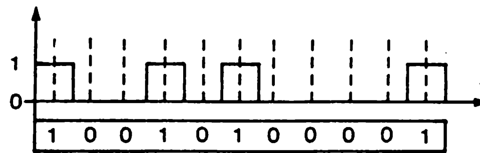


Fig. 1-18. Information série.

- **Transmission parallèle**

  L'information codée sur n créneaux est transmise en délivrant tous les créneaux simultanément sur n fils (fig. 1.19).

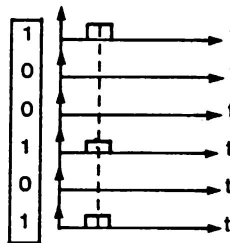


Fig. 1-19. Information parallèle.

10. Minimisation des équations logiques

- **Méthode algébrique**

On utilise les propriétés combinées de la somme et du produit logique. Celles-ci se résument comme suit :

- Les produits de somme logique se développent comme en algèbre ordinaire :

$$x(y+z) = xy + xz$$

- Lorsqu'une somme logique contient un terme et un de ses multiples on peut négliger le multiple :

$$x + xy = x(1+y) = x$$

- Lorsqu'une somme logique contient une quantité et un multiple de son complément on peut faire disparaître le complément :

$$x + \bar{x}y = x + y$$

En effet on peut écrire :

$$x = x + xy \text{ et } x + \bar{x}y = x + y(x + \bar{x}) = x + y$$

- Pour ajouter un produit logique à un terme on ajoute chacun des facteurs au terme et on fait le produit logique des deux sommes logiques obtenues.

$$x + yz = (x + y)(x + z)$$

En effet $x + yz$ peut s'écrire $x + xy + yz$ en remplaçant x par $x + xy$.
De même $x + xy + yz = x + xz + xy + yz$ en remplaçant x par $x + xz$ et en écrivant $x = xx$, on écrit :

$$xx + xz + xy + yz = x(x + z) + y(x + z)$$
$$= (x + z)(x + y)$$

Exemple :

1°) $X = a\bar{c} + b + a\bar{b}c$

$X = a\bar{c} + b + ac$

$X = a(\bar{c} + c) + b = a + b$

2°) $R = ab\bar{c} + abc + a\bar{b}\bar{c}$

$R = ab(\bar{c} + c) + a\bar{b}\bar{c}$

$R = a(b + \bar{b}\bar{c})$

$R = a(b + \bar{c})$

• Utilisation des tableaux de Karnaugh

Nous savons représenter des ensembles a et b inscrits dans un référentiel M. On peut donc définir bon nombre de surfaces parmi lesquelles a, b, ab, $\bar{a}$, $\bar{b}$, etc. (fig. 1.20).

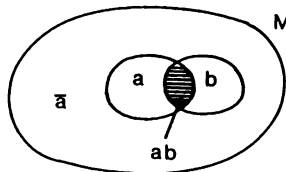


Fig. 1-20. Représentation d'ensembles.

Pour des raisons de commodité, le cercle référentiel M peut se transformer en un carré ou un rectangle qui constitue un nouveau référentiel. Dans ce dernier nous plaçon d'une façon particulière les surfaces a et b (fig. 1-21).

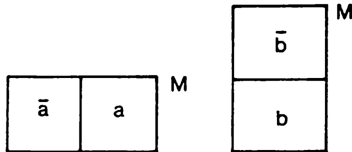


Fig. 1-21. Autre représentation.

Enfin, en regroupant les deux représentations précédentes en une seule, nous définissons quatre surfaces (fig. 1.22).

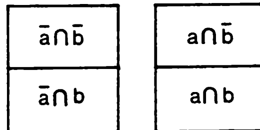


Fig. 1-22. Représentation des intersections.

Pour une meilleure utilisation on présente le tableau de Karnaugh sous la forme suivante (fig. 1.23).

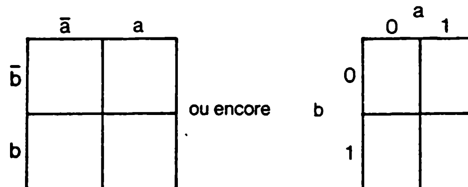


Fig. 1-23. Tableaux de Karnaugh.

Remarque :

On fait correspondre à **a** la valeur 1 et à **ā** la valeur 0, c'est ce que nous avons appelé l'état technologique.

Cas de trois variables (fig. 1.24)

c \ ab	00	01	11	10
0				
1				

ou

bc \ a	0	1
00		
01		
11		
10		

Fig. 1-24. Tableau de Karnaugh à 3 variables.

Exemple d'utilisation

Soit à simplifier $L = a + ab$

On affecte de « 1 » les cases correspondant au fonctionnement du récepteur L. Et de « 0 » les cases correspondant au non-fonctionnement du récepteur L (fig. 1.25).

b \ a (L)	0	1
0	0	1
1	0	1

Fig. 1-25. Remplissage du tableau de Karnaugh

Remarque

On pourra affecter certaines cases de 0 correspondant à une impossibilité technologique ou de fonctionnement pour les combinaisons de variables ainsi repérées.

Pour obtenir l'équation booléenne de L la plus simple, il suffit de faire le regroupement **maximal** de cases affectées de 1.

Ici, $L = a$

Soit à simplifier les équations booléennes suivantes :

$$X = a\bar{c} + b + a\bar{b}c \text{ et } R = ab\bar{c} + abc + a\bar{b}c$$

Les tableaux de Karnaugh relatifs à X et R sont : (fig. 1.26).

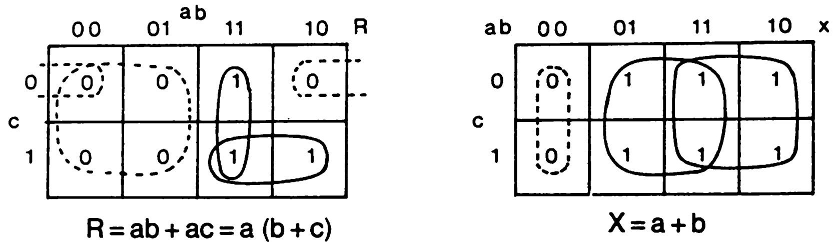


Fig. 1-26. Regroupements et équations relatives dans des tableaux de Karnaugh

Remarque

Si dans les tableaux précédents on faisait un regroupement de « 0 », on obtiendrait le non-fonctionnement des sorties X et R.

Ainsi : $\bar{X} = \bar{a}\,\bar{b}$ et $\bar{R} = \bar{a} + \bar{b}\,\bar{c}$

En considérant les équations de X et $\bar{X}$ et de R et $\bar{R}$, on en déduit les théorèmes dits de **De Morgan.**

- Le complément d'une **somme** logique est égal au **produit logique** des compléments des termes de cette somme.

$$X = a + b \Longrightarrow \bar{X} = \overline{a + b} = \bar{a} \cdot \bar{b}$$

- le complément d'un **produit** logique est égal à la **somme logique** des compléments des termes de ce produit.

$$R = a \cdot (b + c) \Longrightarrow \begin{array}{l} \bar{R} = \overline{a \cdot (b + c)} \\ \bar{R} = \bar{a} + \overline{(b + c)} \\ \bar{R} = \bar{a} + \bar{b} \cdot \bar{c} \end{array}$$

II - Opérateurs logiques fondamentaux

1. Définition

Une équation logique quelconque, quel que soit le nombre de variables, s'écrit seulement avec les opérateurs NON, OU et ET.

Exemple : $L = (a + \bar{b}c)\, d + \bar{e}$

2. Représentation symbolique et table de vérité des opérateurs fondamentaux :

Opérateur NON (fig. 2-1)

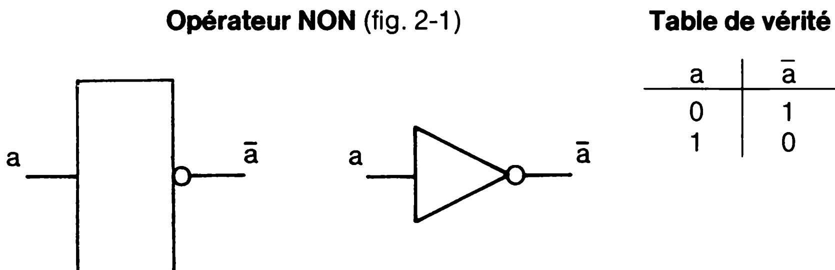
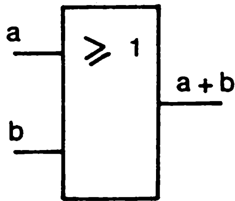


Table de vérité

a	$\bar{a}$
0	1
1	0

Fig. 2-1. Représentation européenne et représentation anglo-saxonne de l'opérateur NON.

Opérateur OU (fig. 2-2)



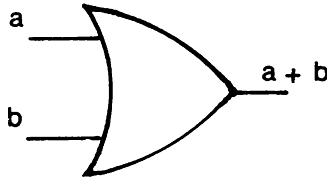
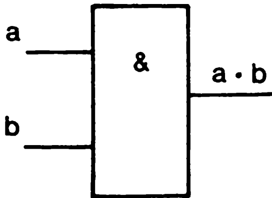


Table de vérité

a b	a + b
0 0	0
0 1	1
1 0	1
1 1	1

Fig. 2-2. Représentation européenne et représentation anglo-saxonne de l'opérateur OU..

Opérateur ET (fig. 2-3)



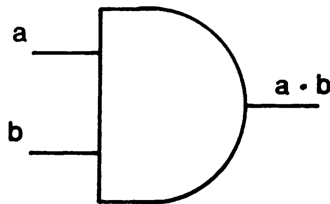


Table de vérité

a b	a • b
0 0	0
0 1	0
1 0	0
1 1	1

Fig. 2-3. Opérateurs ET.

3. Autres opérateurs

• L'opérateur NAND (NON - ET) :

Nous allons voir qu'il est possible de réaliser n'importe laquelle des fonctions logiques avec pour seul opérateur le NAND.

Représentation symbolique (fig. 2 4)

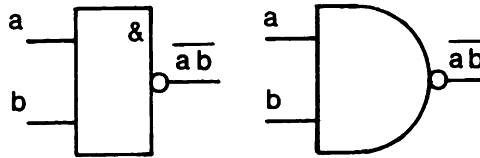
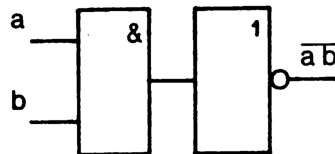


Fig. 2-4. Opérateur NAND.

Table de vérité

a b	$\overline{a b}$
0 0	1
0 1	1
1 0	1
1 1	0

équivalent à



• L'opérateur NOR (NON - OU) :

Il est aussi possible de réaliser n'importe laquelle des fonctions logiques avec pour seule opérateur le NOR.

Représentation symbolique (fig. 2-5)

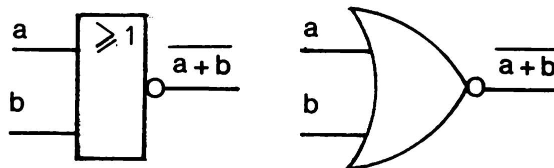
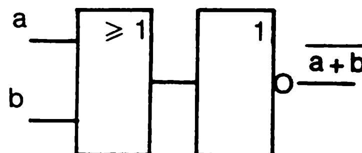


Fig. 2-5. Opérateur NOR.

Table de vérité

a b	$\overline{a + b}$
0 0	1
0 1	1
1 0	1
1 1	0

équivalent à



4. Réalisation des fonctions logiques à l'aide d'opérateurs NAND et NOR

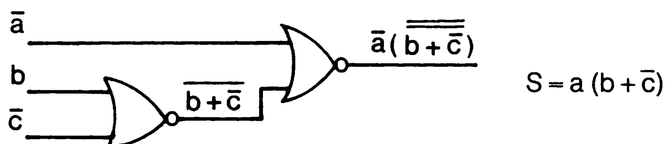
• Méthode

Toujours partir de l'équation finale mise sous la forme d'un produit de sommes si on veut utiliser des opérateurs NOR et somme de produits pour des opérateurs NAND et « remonter » la fonction jusqu'à ce qu'il y ait une seule variable.

Lorsqu'on utilise des opérateurs NOR, si la sortie est une fonction OU l'opérateur de sortie est un NON.

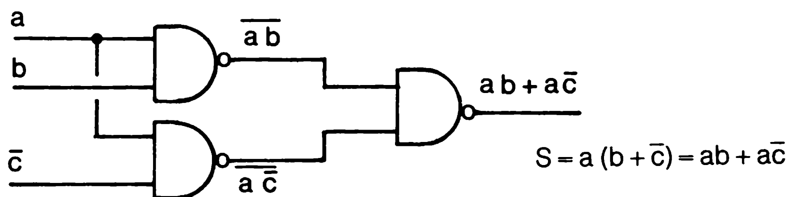
Lorsqu'on utilise des opérateurs NAND si la sortie est une fonction ET l'opérateur de sortie est un NON.

Exemples (fig. 2-6 et 2-7)



$S = a\,(b + \bar{c})$

Fig. 2-6. Schéma à NOR.



$S = a\,(b + \bar{c}) = ab + a\bar{c}$

Fig. 2-7. Schéma à NAND.

Remarques :

Technologiquement, il est aussi facile d'avoir une variable seule ou son complément, il n'est donc pas toujours nécessaire d'utiliser des fonctions NON en final pour complémenter cette variable.

Il ne faut jamais laisser une entrée en l'air dans un circuit logique, pour cela les entrées inutilisées seront reliées à un potentiel ; même si la fonction n'est pas du tout utilisée. (Cas par exemple du boîtier dans lequel existe 4 portes NAND et 2 seulement sont utilisées).

5. Opérateur OU exclusif

L'opérateur « ou exclusif » nous permet dans certaines applications de réaliser des comparaisons entre mots binaires, de complémentations, etc.

Représentation symbolique (fig. 2.8)

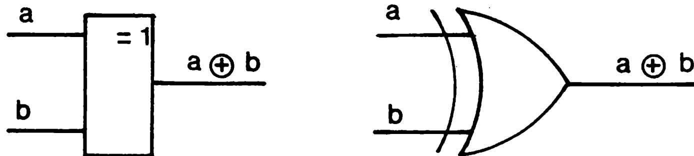
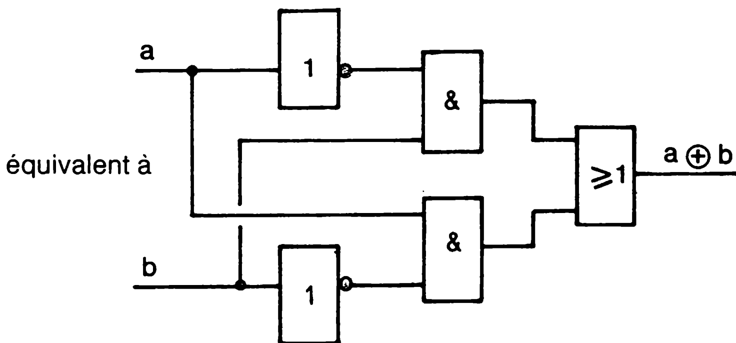


Fig. 2-8. Opérateur ou exclusif.

Table de vérité

a b	a ⊕ b
0 0	0
0 1	1
1 0	1
1 1	0

équivalent à



Remarque : $a \oplus b = a\bar{b} + \bar{a}b$

6. Interface entre la logique à contact et la logique à niveaux

C'est le cas où les informations d'entrée sont produites par des boutons-poussoirs, fin de course, etc.

Exemple : Bouton-poussoir (fig. 2.9)

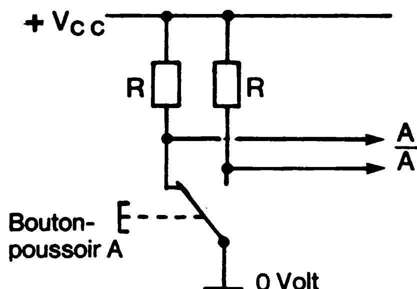


Fig. 2-9. Interface entre logique à contact et logique à niveaux.

Table de vérité

Etat physique	Sortie A	Sortie $\bar{A}$
Pas d'action : 0	0	1
Action : 1	1	0

Avec ce montage les sorties A ou $\bar{A}$ ne pourront prendre que 2 états logiques (0 ou 1) et ne seront jamais « en l'air ». Elles pourront donc « attaquer » les portes logiques.

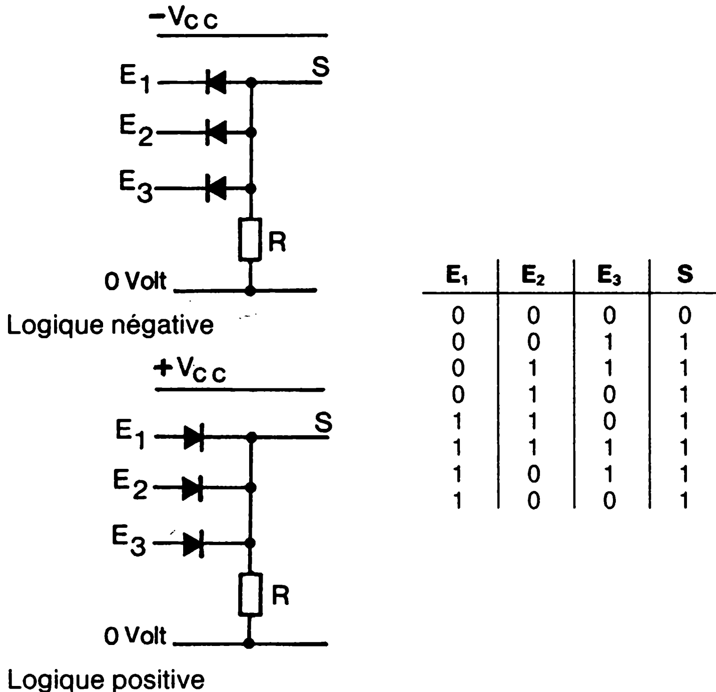
III - Familles logiques

Nous allons, dans ce chapitre, étudier les principales familles technologiques depuis les portes à diodes des années cinquante jusqu'aux circuits actuels pour voir l'évolution technologique qui s'est opérée en 35 ans.

1. Fonctions à diode

Cette technologie est apparue à la fin des années cinquante.

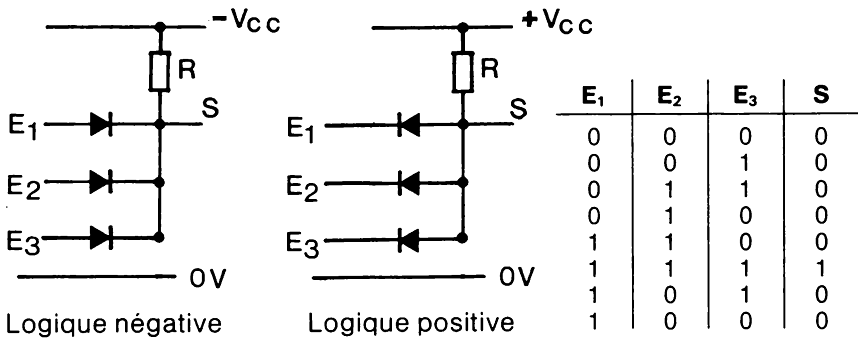
Fonction OU (fig. 3.1)



E_1	E_2	E_3	S
0	0	0	0
0	0	1	1
0	1	1	1
0	1	0	1
1	1	0	1
1	1	1	1
1	0	1	1
1	0	0	1

Fig. 3-1. Porte OU à diodes.

Fonction ET (fig. 3.2)



E_1	E_2	E_3	S
0	0	0	0
0	0	1	0
0	1	1	0
0	1	0	0
1	1	0	0
1	1	1	1
1	0	1	0
1	0	0	0

Fig. 3-2. Porte ET à diodes.

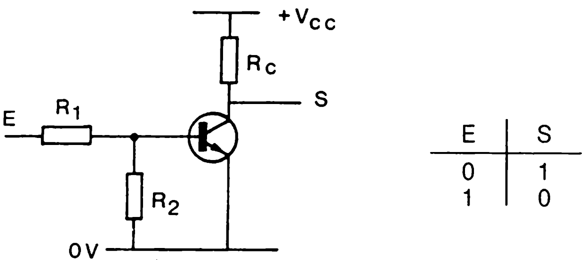
Critiques :

- Dégradation du signal de sortie due à la résistance interne des diodes.
- Dans la fonction ET lorsque l'entrée est à 0 un courant circule, ce qui provoque la chute de tension dans R et S $\neq$ 0.
- Dans la fonction ET la sortie n'est jamais égale à 0 à cause de la chute de tension dans la diode.

2. Fonctions à transistors

(Technologie RTL : Résistor, Transistor, Logic)

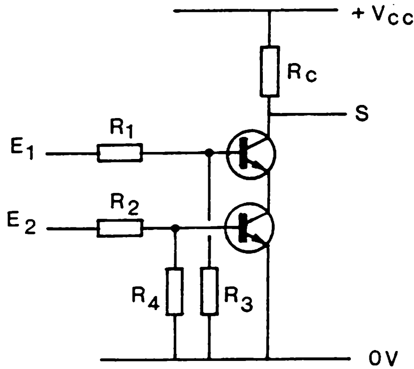
Fonction NON (fig. 3.3)



E	S
0	1
1	0

Fig. 3-3. Porte NON à transistor et résistances.

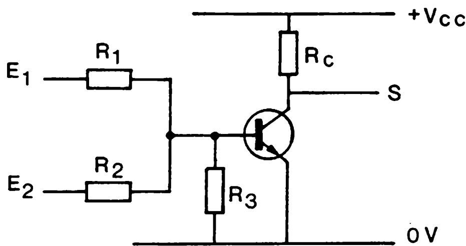
Fonction NAND (fig. 3.4)



E_1	E_2	S
0	0	1
0	1	1
1	0	1
1	1	0

Fig. 3-4. Porte NAND à transistors et résistances.

Fonction NOR (fig. 3-5)



E_1	E_2	S
0	0	1
0	1	0
1	0	0
1	1	0

Fig. 3-5. Porte NOR à transistor et résistances.

Critiques :

- Nombre d'entrées limité (impédances d'entrée variant avec le nombre d'entrées).
- L'impédance de sortie varie pour les états bloqués et saturés. Il résulte de cette différence d'impédance une importante dissymétrie des signaux.

3. Fonctions à diodes et transistors

(Technologique DTL : Diode, Transistor Logic)

Le transistor en sortie régénère le signal.

Fonction NOR (fig. 3-6)

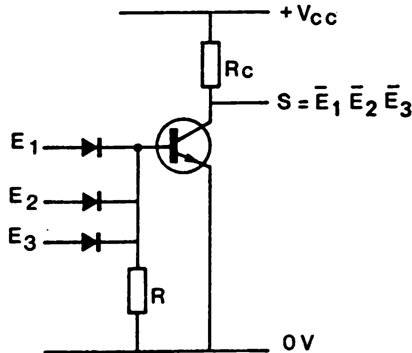


Fig. 3-6. Porte NOR en DTL

Fonction NAND (fig. 3.7)

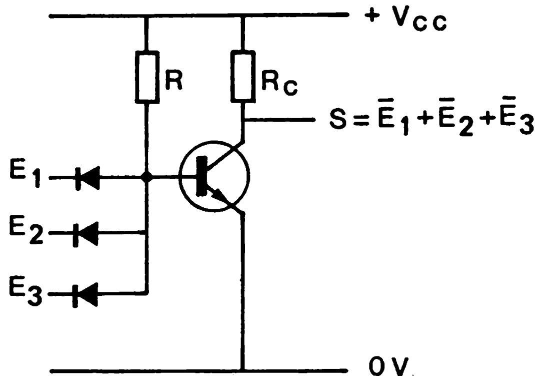


Fig. 3-7. Porte NAND en DTL

Critiques :

La technologie DTL ne présente pas les défauts des portes à diodes (dégradation du signal) car le transistor régénère le signal (bon niveau de sortie).

Remarque :

La technologique DTL était très employée dans les années soixante avant l'apparition de la TTL.

4. Technologie TTL
(Transistor, Transistor Logic)

C'est, avec la technologie CMOS, celle qui propose actuellement le plus grand nombre de circuits. Elle se caractérise par le fait que les circuits d'entrée et de sortie sont réalisés avec des transistors.

• Schéma : Porte TTL NAND standard (série 74) (fig. 3.8)

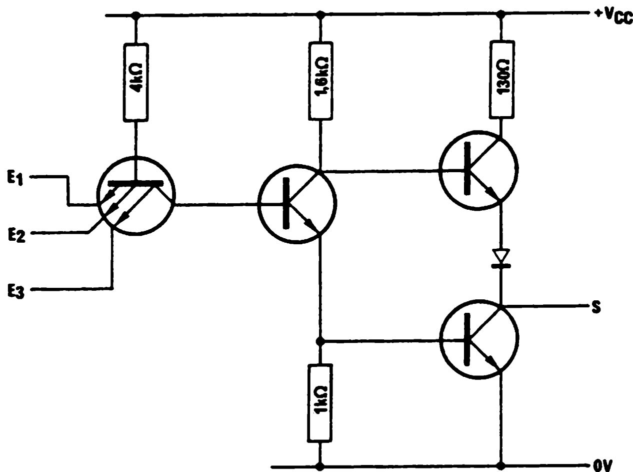


Fig. 3-8. Porte TTL standard.

• Explication :

Considérons le transistor d'entrée multiémetteur comme un ensemble de jonctions PN (fig. 3.9).

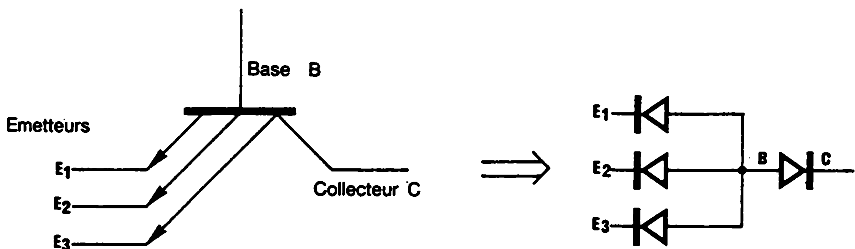


Fig. 3-9. Schéma équivalent du transistor multiémetteur.

La porte NAND fondamentale devient : (fig. 3.10)

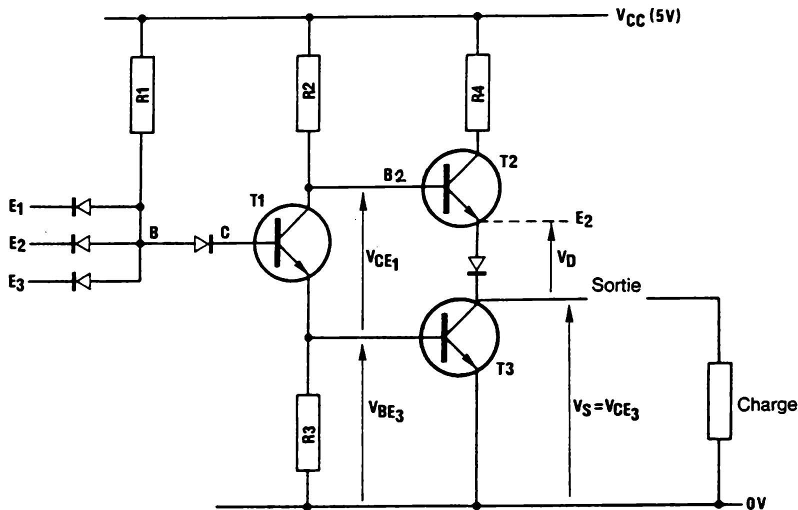


Fig. 3-10. Porte NAND en DTL.

a : Si les entrées E_1, E_2, E_3, sont en l'air ou reliées à + Vcc :

Les transistors T_1 et T_3 sont saturés.
On peut écrire $V_S = V_{CE_3}$ Sat soit $\simeq$ 0V
Le potentiel B_2 s'écrit :
$V_{B_2} = V_{CE_1}$ Sat + V_{BE_3} Sat
Le potentiel E_2 s'écrit :
$V_{E_2} = V_{CE_3}$ Sat + VD
Et comme $V_D \simeq V_{BE_3}$ Sat, on dira que :
$\mathbf{V_{B_2} = V_{E_2}}$ donc le transistor T_2 est bloqué.

Remarque :

Le transistor T_3 laisse passer un courant de 16 mA. Chaque entrée consommant un courant de 1,6 mA (au niveau 0), la porte a une sortance de 10.

b : Si une entrée au moins est à l'état bas :

Le transistor T_1 est bloqué, ce qui bloque T_3
Le transistor T_2 est saturé. On peut écrire :
VS $\simeq$ Vcc

Remarque :

Le courant de sortie est de 400 µA, il est suffisant pour attaquer 10 portes semblables puisque chaque entrée au niveau haut consomme 40 µA environ (sortance de 10).

Résistance de sortie :

10 Ω ⩽ Résistance de sortie ⩽ 20 Ω à l'état bas.
Rs ≃ 70 Ω à l'état haut.

• Critiques :

- Toutes les entrées sont réalisées par un transistor multiémetteur.
- L'impédance de sortie reste faible quelque soit le niveau logique.

• Remarque :

Il y a dans la technologie TTL des séries qui se caractérisent par des différences au niveau des consommations et des vitesses de propagation du signal.

Ces différentes séries se reconnaissent par la référence qu'elles portent sur le boîtier :

- 54 / 74 série normale :
  Retard de 10 ns et une consommation de 10 mW pour une porte fondamentale.
- 54 H / 74 H série grande rapidité :
  Retard de 6 ns et une consommation de 22 mW.
- 54 J / 74 L série faible consommation :
  Retard de 33 ns et une consommation de 1 mW.
- 54 S / 74 S série rapide (Transistor SCHOTTKY) :
  Retard de 3 ns et une consommation de 20 mW par porte.
- 54 LS / 74 LS série rapide et faible consommation (Transistor SCHOTTKY) :
  Retard de 9,5 ns et une consommation de 2 mW par porte.

L'utilisateur devra connaître ces caractéristiques avant de choisir une série ou une autre pour réaliser son système.

La référence 54 désigne la technologie TTL gamme « militaire » : température de fonctionnement comprise entre – 55° et + 125°C.

Et la référence 74 désigne la technologie TTL gamme « industrielle » : température de fonctionnement comprise entre 0° et 70°C.

En stockage les circuits intégrés peuvent être placés à une température plus élevée qui est donnée par le constructeur.

• Conclusions :

Les avantages de la famille TTL qui propose actuellement le plus grand choix de circuits sont :

- la fiabilité,
- le prix de revient très bas,
- la simplicité d'utilisation.

Il faut toutefois noter sa relative immunité aux bruits et sa consommation relativement élevée, par rapport aux technologies utilisant des transistors à effet de champ comme la CMOS.

5. Technologie CMOS

Le principe de cette technologie réside dans l'utilisation de deux transistors à effet de champ complémentaire (canal P et canal N) pour la réalisation de la porte fondamentale (fig. 3.11).

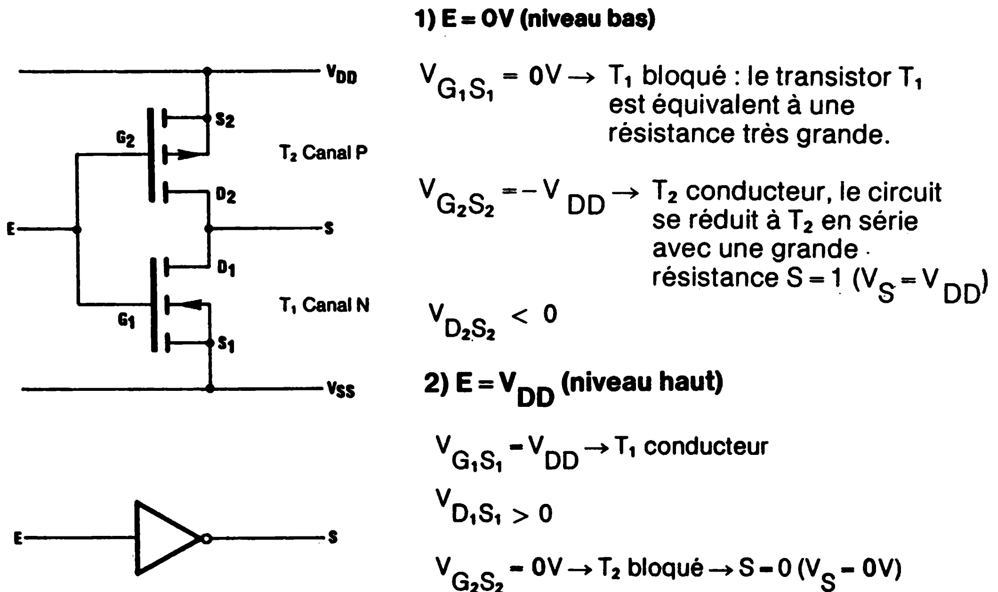


Fig. 3-11. Inverseur CMOS.

Critiques :

- Par l'utilisation de transistors à effet de champ, les impédances d'entrée sont très élevées et une porte consomme de 1 à 150 nW.
- Les temps de propagation qui étaient relativement longs par rapport à la TTL semblent, avec les progrès, diminuer considérablement.
- Les constructeurs nous permettent d'alimenter les circuits CMOS avec une tension comprise entre 3 V et 18 V. Ce qui augmente encore leur immunité aux bruits parasites.

Les fabricants nous proposent un grand choix de circuits dans cette technologie : série 4000 ou 74 C dont le brochage est identique à la série 74 (TTL).

6. Technologie MOS

Les progrès sur l'intégration des circuits ont permis de développer ces dernières années des circuits encore plus complexes utilisant des transistors MOS (plus faciles à réaliser).
Ce sont surtout les microprocesseurs qui en ont bénéficié, d'abord en MOS canal P et ensuite en canal N.

7. Les logiques non saturées

Les logiques non saturées comme l'ECL (Emitter Coupled Logic) utilisent les transistors non pas en saturé-bloqué (transistor en commutation) mais en fonctionnement autour d'un point de repos (en linéaire). Ceci à l'avantage d'augmenter la vitesse de propagation (moins de la ns pour la porte fondamentale).
Des familles de circuits classiques sont commercialisées, telles les séries ECL 10 K et 100 K de RTC-Compelec, riches de plus de 60 fonctions logiques.

Conclusion

Il existe d'autres familles technologiques moins employées car les plus usitées sont la TTL et la CMOS pour la grande variété de circuits qu'elles présentent, et la MOS pour les circuits de type LSI (Large Scale Intégration) ou VLSI (Very Large Scale Intégration). Les fabricants s'efforcent d'améliorer la fiabilité et le niveau d'intégration. Ce qui se traduit par une utilisation de plus en plus grande de ces circuits dans tous les domaines industriels d'où une baisse progressive des prix de revient.

IV - Systèmes de numération

1. Numération binaire

On appelle système de numération une technique qui permet d'écrire tout entier naturel.

Tout système est caractérisé par le nombre de chiffres qu'il utilise. Ce nombre s'appelle la base du système.

Si b est la base du système, tout entier s'écrira :

$A_b = r_{n-1} \quad b^{n-1} + ... + r_2 \quad b^2 + r_1 \quad b + r_0$

On utilise couramment 4 bases de numération dans les systèmes logiques.

- Base 2 (symbole 0 et 1)
- Base 8 (symbole 0.1.2.3.4.5.6.7)
- Base 10 (symbole 0.1.2.3.4.5.6.7.8.9)
- Base 16 (symbole 0.1.2.3.4.5.6.7.8.9.A.B.C.D.E.F)

Un nombre entier N s'écrira en base 2.

$N_2 = r_{n-1} 2^{n-1} \quad +... + r_2 2^2 + r_1 \quad 2 + r_0$

Exemple

$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2 + 1$

Tableau de conversion décimal - binaire

Décimal	Binaire
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001

Passage du décimal au binaire

On utilise la méthode des divisions successives qui consiste à diviser le nombre décimal par 2 et successivement les quotients obtenus jusqu'à ce que le dernier quotient soit égal à 1.

On relie ensuite les restes (0 ou 1) des divisions successives du dernier au premier pour obtenir le nombre converti en binaire.

Exemple

– soit à convertir 27_{10} en binaire.
On fait les divisions suivantes :

```
27 | 2
 1 | 13 | 2
      1 |  6 | 2
           0 | 3 | 2
               1 | 1
```

et le résultat est : $27_{10} = 11011_2$

Passage du binaire au décimal

On convertit le nombre binaire en décimal par sommation des puissances de 2.

Exemple - $11011_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2 + 1 = 27_{10}$.

Il est fastidieux d'opérer, pour l'utilisateur, sur les nombres binaires (risques d'erreurs, mot souvent trop souvent longs). On préfère représenter ces nombres binaires en **octal** ou en **hexadécimal** qui conservent la pondération du mot.

2. Représentation octale

Par divisions successives par 8, on remarque qu'un nombre binaire peut s'écrire facilement en octal en séparant par tranche de trois éléments (bits) (converti en octal) le nombre écrit en binaire.

Exemple

$N_2 = 1001001011_2$

On sépare ce nombre en tranche de 3 bits.

```
N₂ =  001  001  001  011
       ↓    ↓    ↓    ↓
       1    1    1    3
```

Donc $1001001011_2 = 1113_8$

La conversion inverse est aussi simple.

6	3	5
↓	↓	↓
110	011	101

Donc $6\ 3\ 5_8 = 110011101_2$

Pondération

Exemple

$N_2 = \underbrace{101}\ \underbrace{011} = 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 43_{10}$

$N_8 = \ 5 \quad 3_8 = 5 \times 8^1 + 3 \times 8^0 = 43_{10}$

3. Représentation hexadécimale

Les mêmes méthodes peuvent être employées pour convertir un nombre binaire en nombre hexadécimal et inversement.

Transformation binaire → hexadécimale

Exemple : 101100101101_2

On sépare ce nombre en tranches de 4 bits.

1011	0010	1101
↓	↓	↓
B	2	D

Transformation hexadécimale → binaire

Exemple : E 8 3

E	8	3
↓	↓	↓
1110	1000	0011

Rappel :

Correspondances décimale, binaire, hexadécimale.

décimale	binaire	hexadécimale
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

Pondération

$$N_2 = \underbrace{1011}_{B}$$

$$\underbrace{0111}_{7} = \underbrace{1 \times 2^7}_{128} + \underbrace{0 \times 2^6}_{0} + \underbrace{1 \times 2^5}_{32} + \underbrace{1 \times 2^4}_{16} + \underbrace{0 \times 2^3}_{0} + \underbrace{1 \times 2^2}_{4} + \underbrace{1 \times 2^1}_{2} + \underbrace{1 \times 2^0}_{1} = 183_{10}$$

$N_{16} = B\,7 = 11 \times 16^1 + 7 \times 16^0 = 183_{10}$

Nota :

Dans les microprocesseurs 8 bits (c'est-à-dire qu'ils traitent des données de 8 bits : 1 octet), il est classique de dimensionner le bus d'adresses sur 16 bits (ce qui leur permet une capacité d'adressage de 2^{16} adresses différentes pour les programmes).

Il est commode de représenter, pour la programmation dite en langage machine, les adresses et les données (16 et 8 bits) en hexadécimal.

Ainsi par exemple :

OE00	C 4
Adresse	Donnée

4. Représentation BCD *(binary coded decimal)*

Travailler sur des nombres binaires purs (représentés en octal ou en hexadécimal) est intéressant dans les calculateurs, car ces nombres sont pondérés, mais, lorsqu'on veut une image rapide de l'équivalent décimal, on est amené à effectuer un transcodage long et fastidieux.

Il est plus commode dans certaines applications comme par exemple l'affichage en décimal du contenu de compteurs d'utiliser la représentation BCD.

Celle-ci consiste à remplacer les dix chiffres significatifs de la numération décimale par un mot de 4 bits, dont la progression est la numération binaire pure.

Exemple :	1	9	7	3	en décimal
	↓	↓	↓	↓	
devient :	0001	1001	0111	0011	en BCD
Et le nombre	1000	0001	0100	0000	en BCD
	↓	↓	↓	↓	
devient	8	1	4	0	en décimal

Exemple d'utilisation (fig. 4.1)

Les 4 bits de sortie de chaque décade présentent une image du nombre d'impulsions reçues sur l'entrée de comptage avec :

$$0000 \leqslant N \leqslant 1001$$

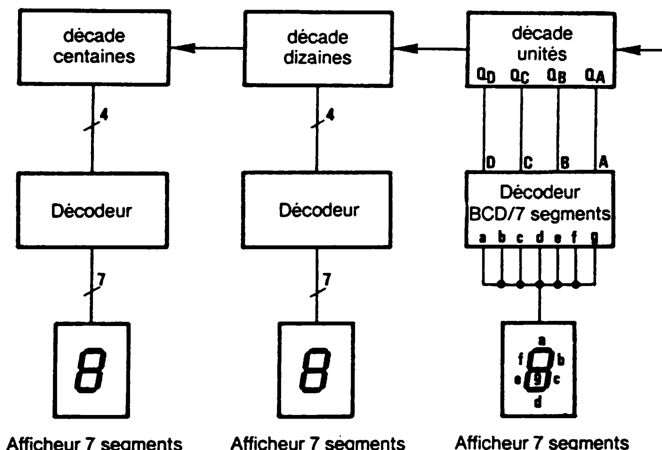


Fig. 4-1. Principe comptage-décodage-affichage 3 digits.

5. Codes non pondérés

Ces différents codes sont des codes binaires mais ils ne vérifient pas :

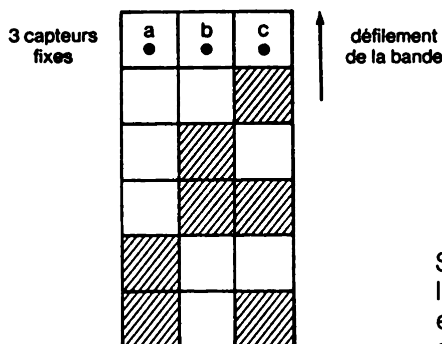
$$\text{terme } n = \text{élément } a \times 2^p$$

S'ils ne sont pas pondérés, ils présentent d'autres avantages dans certaines utilisations particulières.

• Binaire réfléchi : (code GRAY)

Principe :

Imaginons une bande sur laquelle les hachures représentent les 1 et 3 capteurs alignés a, b, c, qui reconnaîtraient les 0 et les 1 (fig. 4.2).



Si les capteurs sont mal alignés lorsqu'on passe de 1 à 2 par exemple, ces capteurs risquent de lire :

001		001
↓		↓
000	ou	011
↓		↓
010		010

Fig. 4-2. Code GRAY.

Conclusion :

Avec le code binaire pur, si les capteurs sont mal alignés, ils indiquent un signal temporaire faux ; on imagine donc un autre code qui présente la particularité de n'avoir jamais qu'un seul chiffre qui change quand on passe d'un nombre au suivant.

Table de correspondance

décimal	binaire pur	binaire réfléchi
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100

décimal	binaire pur	binaire réfléchi
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Relation en un nombre binaire pur et binaire réfléchi

On peut vérifier que

$$n = \left| \frac{N \oplus 2N}{2} \right|$$ ou n nombre binaire réfléchi
N nombre binaire pur

Remarque :

C'est le code binaire réfléchi que nous utilisons dans les tableaux de Karnaugh.

• Codes à excès de trois ou de STIBITZ

C'est le code BCD dans lequel on ajoute 3.

Exemple : $4_{10} = 0100_{BCD} = 0111_{\text{Excès de 3}}$

B (10)	Excès de trois
0	0011
1	0100
2	0101
3	0110
4	0111
5	1000
6	1001
7	1010
8	1011
9	1100

Critiques :

Certaines machines calculent dans ce code car les sommes à retenue sont facilitées. Il est autocomplémentaire, il suffit en effet d'inverser tous les bits du nombre pour obtenir son complément à 9.

4 —— Complément ——→ 5

0111   1000

6. Autres codes

• Code AIKEN :

Ce code est pondéré (1, 2, 4, 2) et autocomplémentaire.

Décimal	**A_3** **2**	**A_2** **4**	**A_1** **2**	**A_0** **1**
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	0	1
8	1	1	1	0
9	1	1	1	1

• (Code ASCII : American Standard Communication Informations Interchange)

L'information est représentée par 7 bits avec éventuellement un 8e bit de parité (bit de contrôle entre émission et réception).
Ce code est couramment employé entre ordinateurs et périphériques comme claviers, imprimantes, etc.

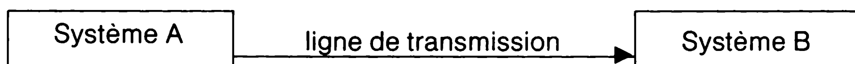


Tableau de correspondance ASCII - hexadécimal (fig. 4-3)

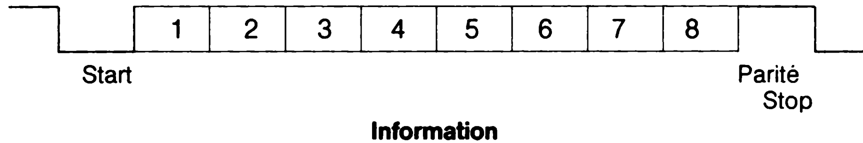
Car	Dec	Hex	Car	Dec	Hex	Car	Dec	Hex	Car	Dec	Hex
NUL	00	00									
SOH	01	01	!	33	21	A	65	41	a	97	61
STX	02	02	”	34	22	B	66	42	b	98	62
ETX	03	03	#	35	23	C	67	43	c	99	63
EOT	04	04	$	36	24	D	68	44	d	100	64
ENQ	05	05	%	37	25	E	69	45	e	101	65
ACK	06	06	&	38	26	F	70	46	f	102	66
BEL	07	07	’	39	27	G	71	47	g	103	67
BS	08	08	(	40	28	H	72	48	h	104	68
HT	09	09	)	41	29	I	73	49	i	105	69
LF	10	0A	*	42	2A	J	74	4A	j	106	6A
VT	11	0B	+	43	2B	K	75	4B	k	107	6B
FF	12	0C	,	44	2C	L	76	4C	l	108	6C
CR	13	0D	–	45	2D	M	77	4D	m	109	6D
SO	14	0E	.	46	2E	N	78	4E	n	110	6E
SI	15	0F	/	47	2F	O	79	4F	o	111	6F
DLE	16	10	0	48	30	P	80	50	p	112	70
DC1	17	11	1	49	31	Q	81	51	q	113	71
DCR	18	12	2	50	32	R	82	52	r	114	72
DC3	19	13	3	51	33	S	83	53	s	115	73
DC4	20	14	4	52	34	T	84	54	t	116	74
NAK	21	15	5	53	35	U	85	55	u	117	75
SYN	22	16	6	54	36	V	86	56	v	118	76
ETB	23	17	7	55	37	W	87	57	w	119	77
CAN	24	18	8	56	38	X	88	58	x	120	78
EM	25	19	9	57	39	Y	89	59	y	121	79
SUB	26	1A	:	58	3A	Z	90	5A	z	122	7A
ESC	27	1B	;	59	3B	[	91	5B		123	7B
FS	28	1C	<	60	3C	\	92	5C		124	7C
GS	29	1D	=	61	3D	]	93	5D	ALT	125	7D
RS	30	1E	>	62	3E	↑	94	5E	ESC	126	7E
US	31	1F	?	63	3F	←	95	5F	DEL	127	7F
SP	32	20	@	64	40		96	60			

Fig. 4-3. Code ASCII.

Les 7 bits sont représentés par la valeur décimale et hexadécimale correspondante.

Format du mot :

L'information utile est encadrée par des signaux de start et de stop.



Nota :

On remarque que l'on peut coder tous les chiffres, les caractères spéciaux, les lettres majuscules ou minuscules ainsi que les commandes de contrôle.

7. Décodage

• Principe :

Dans un système logique les informations sont traitées en binaire pur, mais celles-ci « entrent » et « sortent » du calculateur par des périphériques qui transmettent, visualisent, perforent, impriment, etc. Ces périphériques utilisent le plus souvent d'autres codes.

Il est donc nécessaire de pouvoir passer d'un code à l'autre à l'aide de circuits combinatoires.

Exemple - BCD ⟶ Code à excès de 3.

On devra trouver les équations :

$$a = f_1\ (A, B, C, D)$$
$$b = f_2\ (A, B, C, D)$$
$$c = f_3\ (A, B, C, D)$$
$$d = f_4\ (A, B, C, D)$$

Méthode :

On cherche à minimiser les équations à l'aide de 4 tableaux de Karnaugh relatifs à a, b, c, d, suivant la correspondance de la table de vérité.

Table de vérité

BCD				Excès de 3			
A	C	B	A	d	c	b	a
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0

Tableaux de Karnaugh correspondants et équations des sorties (fig. 4.4)

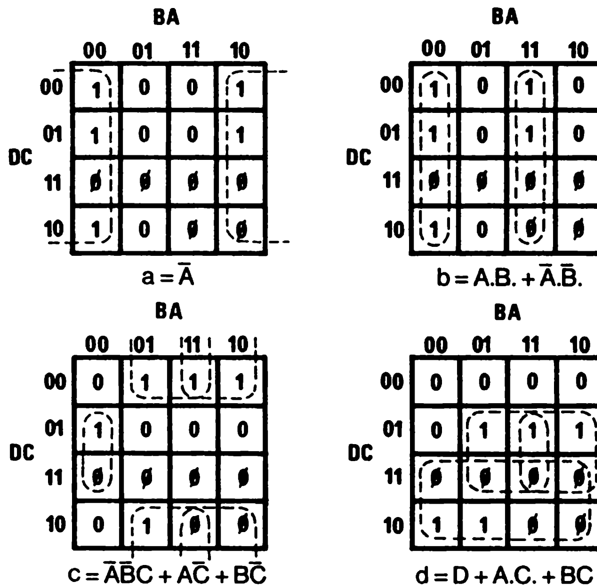


Fig. 4-4.

- Exercice :

Décodeur pour commande visualisation sept segments

Principe : (fig. 4.5)

L'afficheur est composé de sept segments, chaque segment est une diode électroluminescente (LED) ou un élément à cristal liquide (LCD).

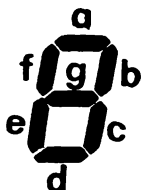


Fig. 4-5. Désignation des segments.

Exercice :

Réaliser le schéma logique pour le segment **a** du décodeur à l'aide d'opérateurs NAND, NOR et inverseur permettant l'affichage sur sept segments des chiffres 0 à 9 selon le modèle suivant, à partir du code BCD (fig. 4-6).

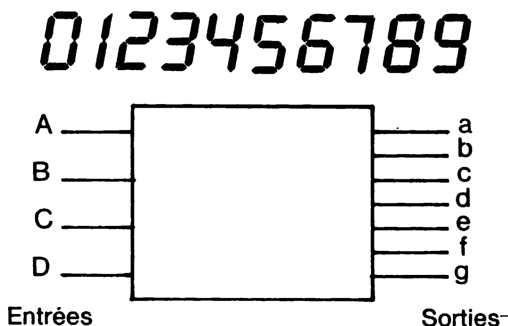


Fig. 4-6. Représentation des chiffres.

Table de vérité

Entrées	Sorties	Affichage
D C B A	abcdefg	
0 0 0 0	1111110	0
1 0 0 1	1111011	9
1 0 1 0	0000000	
Rq		
1 1 1 1	0000000	

Remarque :

Les combinaisons (ABCD) non-utilisées par le décimal devront « éteindre » l'afficheur.

Solution :

Tableau de Karnaugh (fig. 4-7)

DC \ BA	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	0	0	0	0
10	1	1	0	0

Fig. 4-7. Tableau de Karnaugh relatif au segment a.

Equation logique : $a = B\overline{D} + A\overline{D}C + \overline{B}D\overline{C} + \overline{A}\overline{B}\overline{C}$

$a = \overline{D}\,(B + AC) + \overline{B}\overline{C}\,(D + \overline{A})$

Schéma logique (fig. 4-8)

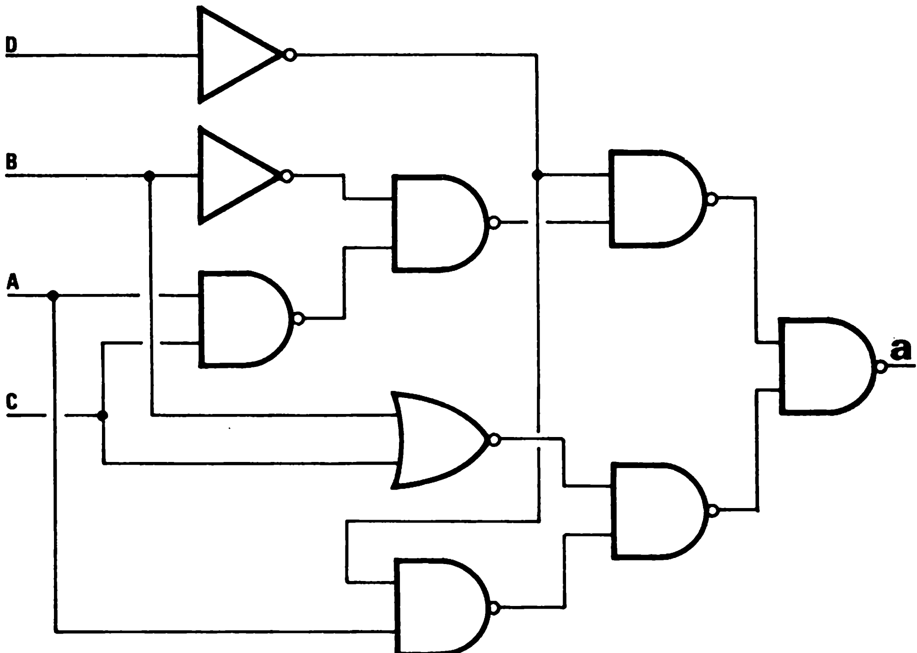
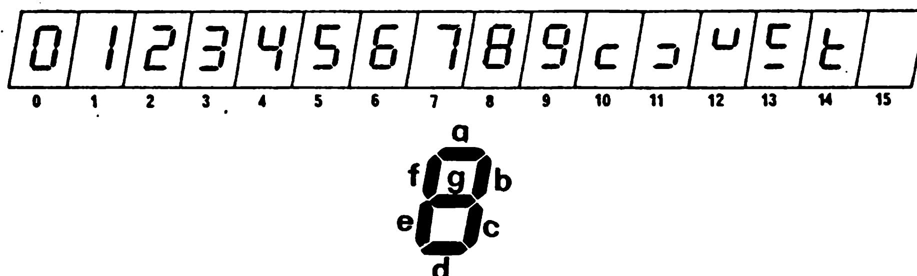


Fig. 4-8. Schéma logique relatif au segment a.

Exemple d'un décodeur intégré (fig. 4-9)



Decimal or Function	Entrée / Input						BI/RBO (1)	Sortie / Output						
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g
0	H	H	L	L	L	L	H	L	L	L	L	L	L	H
1	H	X	L	L	L	H	H	H	L	L	H	H	H	H
2	H	X	L	L	H	L	H	L	L	H	L	L	H	L
3	H	X	L	L	H	H	H	L	L	L	L	H	H	L
4	H	X	L	H	L	L	H	H	L	L	H	H	L	L
5	H	X	L	H	L	H	H	L	H	L	L	H	L	L
6	H	X	L	H	H	L	H	H	H	L	L	L	L	L
7	H	X	L	H	H	H	H	L	L	L	H	H	H	H
8	H	X	H	L	L	L	H	L	L	L	L	L	L	L
9	H	X	H	L	L	H	H	L	L	L	H	H	L	L
10	H	X	H	L	H	L	H	H	H	H	L	L	H	L
11	H	X	H	L	H	H	H	H	H	L	L	H	H	L
12	H	X	H	H	L	L	H	H	L	H	H	H	L	L
13	H	X	H	H	L	H	H	L	H	H	L	H	L	L
14	H	X	H	H	H	L	H	H	H	H	L	L	L	L
15	H	X	H	H	H	H	H	H	H	H	H	H	H	H
BI	X	X	X	X	X	X	L	H	H	H	H	H	H	H
RBI	H	L	L	L	L	L	L	H	H	H	H	H	H	H
LT	L	X	X	X	X	X	H	L	L	L	L	L	L	L

H : niveau haut (High level) L : niveau bas (Low level) X : Indifférent

LT : entrée de test de l'afficheur (Lamp test input)

RBI : entrée d'extinction de l'afficheur (Ripple blanking input)

BI/RBO : ligne de commande d'extinction de l'afficheur (Blanking input or ripple blanking output)

Note : les broches RBI et RI/RBO permettent l'extinction de l'afficheur qui représente un 0 non significatif d'un nombre (exemple : affichage sur une calculatrice).

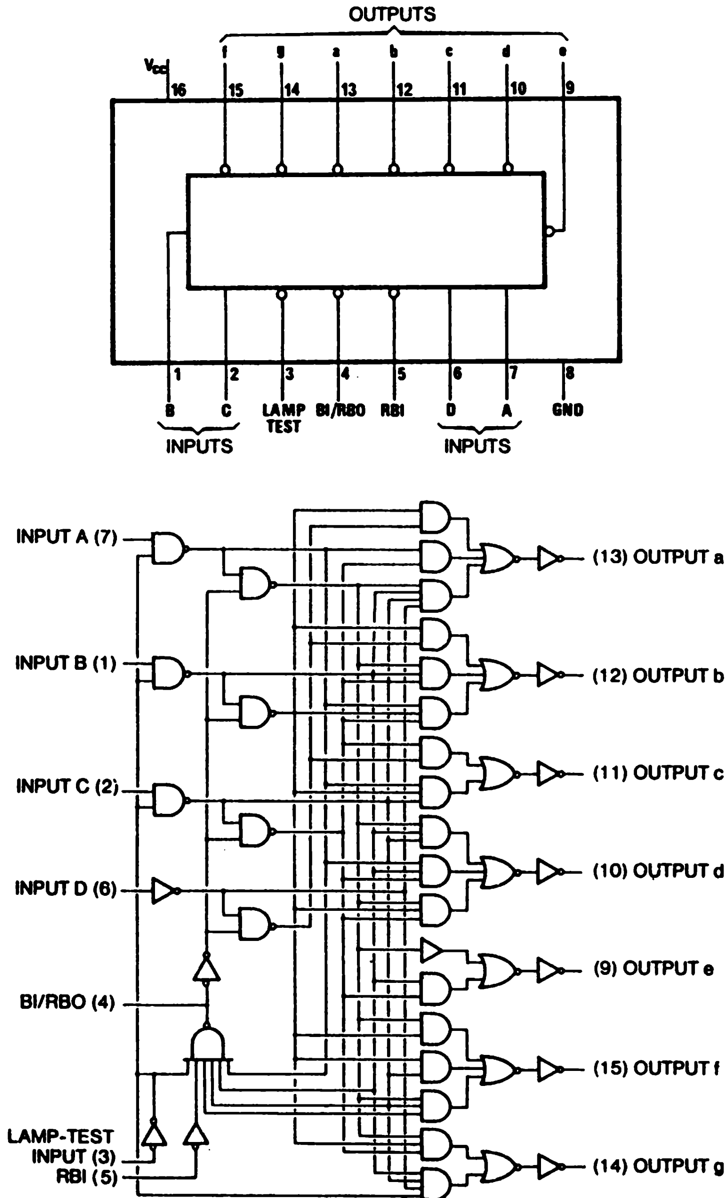


Fig. 4-9. Le décodeur BCD/7 segments 7446.

V - Exemples de circuits intégrés fondamentaux

1. Portes élémentaires

- Portes NON : (fig. 5.1)

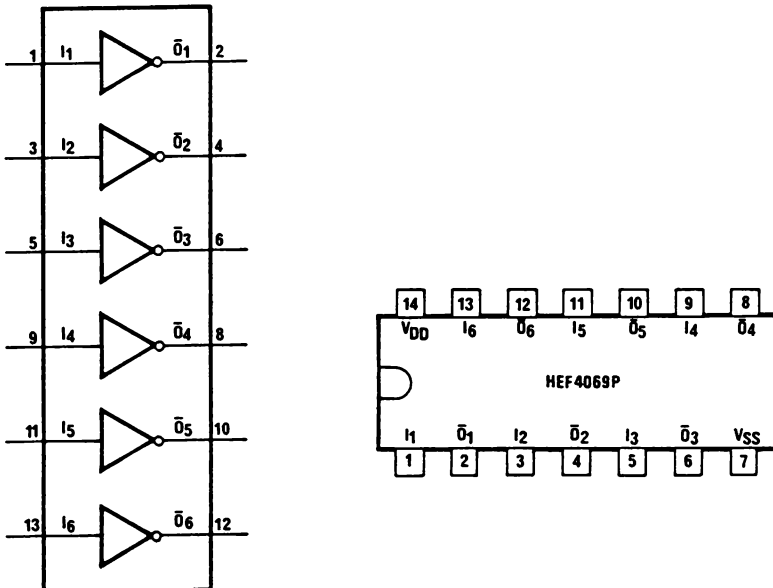


Fig. 5-1. Représentation et brochage du 4069.

• Portes ET : (fig. 5.2)



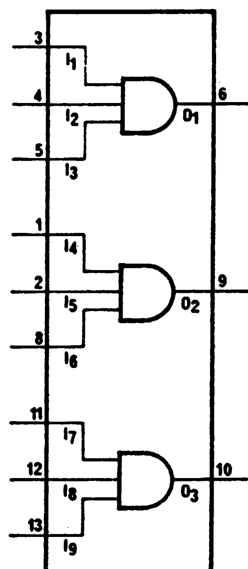
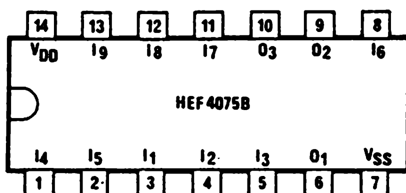


Fig. 5-2. Représentation et brochage du 4073.

• Portes OU : (fig. 5.3)



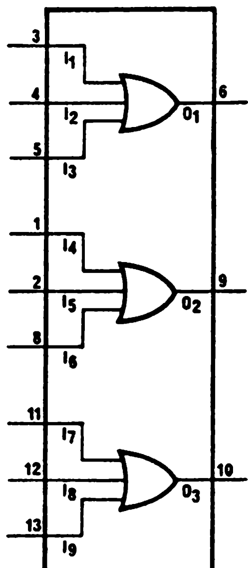
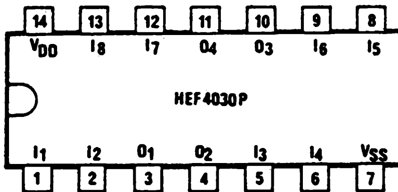


Fig. 5-3. Représentation et brochage du 4075.

• Portes OU exclusifs : (fig. 5.4)



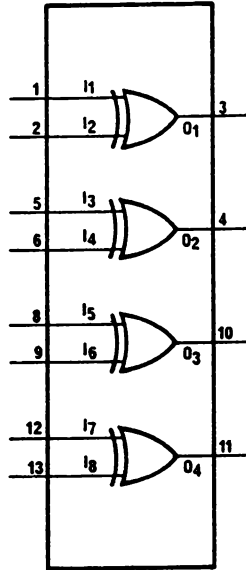
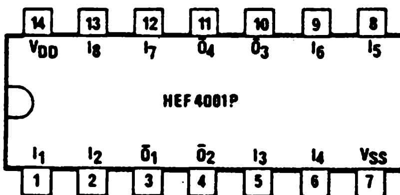
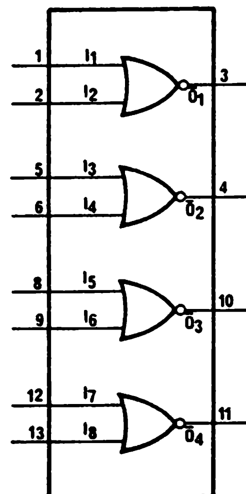
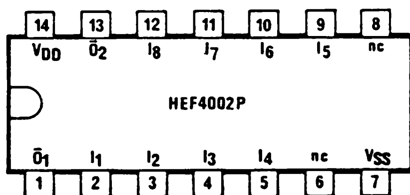


Fig. 5-4. Représentation et brochage du 4030.

• Portes NOR : (fig. 5.5)







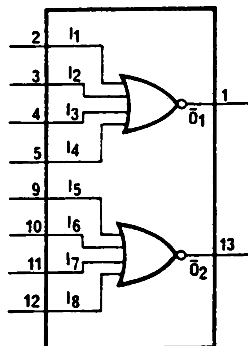
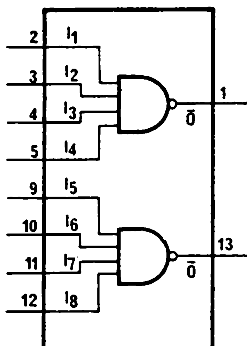
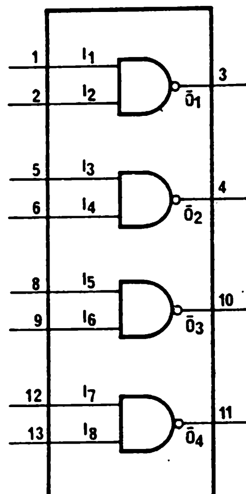
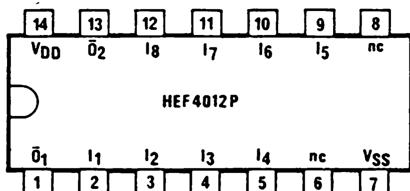


Fig. 5-5. Exemples de circuits NOR.

• Portes NAND : (fig. 5.6)







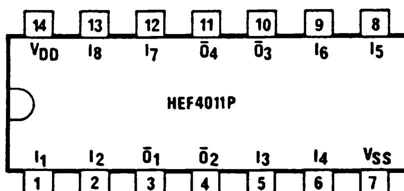


Fig. 5-6 Exemples de circuits NAND.

2. Remarques sur les C.I. logiques

Le brochage de ces portes nous montre que ces circuits sont « encapsulés » dans des boîtiers de type DIL (Dual In Line), qui sont les boîtiers les plus utilisés. On trouve des boîtiers de 8 à 64 broches correspondant à tout l'arsenal des circuits intégrés existants.

Nous allons maintenant étudier des systèmes logiques constitués avec ces circuits de base.

Nous allons voir que nous pouvons réaliser n'importe quel circuit par l'assemblage des circuits de base du système logique.

Sur le plan électronique, les circuits de base peuvent être directement interconnectés entre eux (selon certaines règles concernant le nombre d'entrées maximum connectables à une sortie) si ces circuits appartiennent à une même famille logique.

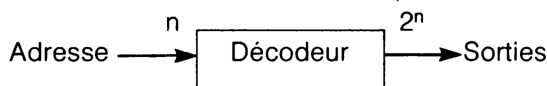
Nota :

On pourra toutefois « passer » d'une famille logique à une autre, à condition d'insérer entre les deux familles une **interface** d'adaptation (voir chapitre XI).

3. Exemples de circuits combinatoires

• Décodeur binaire :

Ce circuit possède n entrées et 2^n sorties. L'entrée n est dite entrée « adresses ».



Une seule sortie est active à la fois : 0 ou 1 suivant le circuit.

Ce circuit permet de « démultiplexer » les n fils d'adresses, c'est-à-dire de sélectionner une sortie parmi 2^n.

Exemple :

Décodeur 2 entrées 4 sorties
1 sortie active à 1 parmi 4

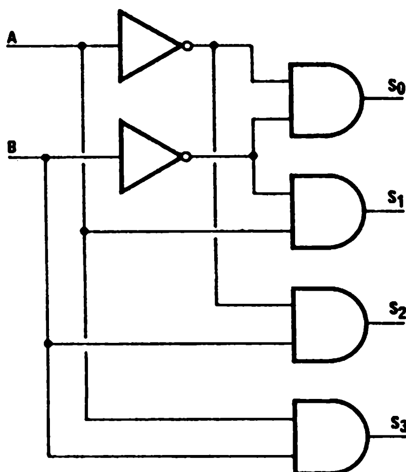
Entrées adresses		Sorties			
B	A	S_0	S_1	S_2	S_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Exercice :

Donner les équations logiques de S_0, S_1, S_2 et S_3, et faire le schéma correspondant.

Solution :

$S_0 = \bar{A}\bar{B}$
$S_1 = A\bar{B}$
$S_2 = \bar{A}B$
$S_3 = AB$



Exemple d'application

Découpage d'une page mémoire de 4 096 adresses différentes (2^{12} combinaisons : bits A_0 à A_{11}) en 4 « sous pages » de 1 024 chacune (2^{10} combinaisons) afin de commander par exemple 4 boîtiers mémoire de 1 024 « places » chacun (fig. 5.7).

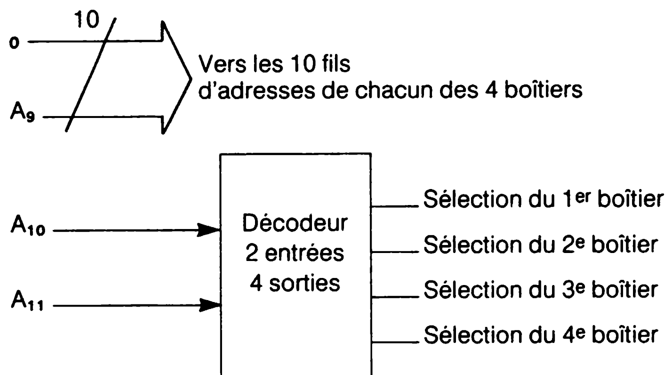


Fig. 5-7. Division d'une page de 4 K en 4 sous-pages de 1 K.

Exemples de décodeurs

– TTL : 74154 (4 vers 16)

Schéma symbolique du 74154 (fig. 5.8)

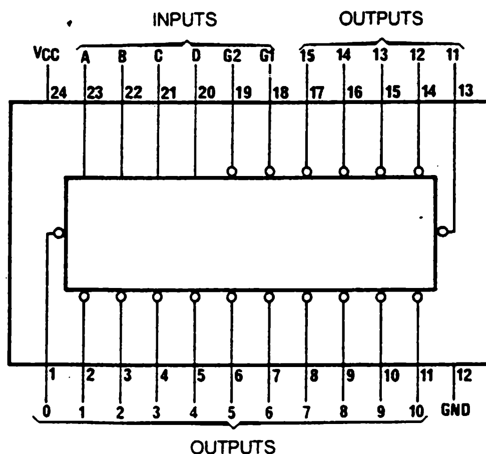


Fig. 5-8.

Table de vérité du décodeur 74154 (fig. 5.9)

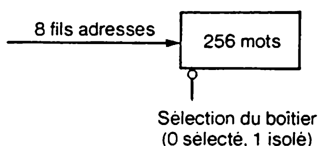
Inputs						Outputs															
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

Fig. 5-9.

Exercice :

Réaliser avec le décodeur 74154 le démultiplexage des 4 bits de poids forts d'une adresse de 12 bits d'un système à microprocesseurs, afin de découper la page de 12 adresses en 16 sous-pages de 256 mots. Ceci aura pour but de commander 16 boîtiers mémoires identiques de 256 mots chacun.

Schéma d'un boîtier



Solution (principe) (fig. 5.10)

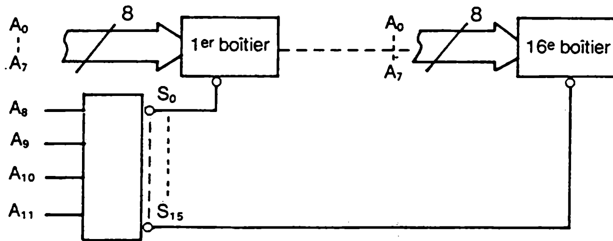


Fig. 5-10. Découpage d'une page de 4 K adresses en 16 sous-pages de 256 adresses.

• Multiplexeurs :

Ce circuit comporte 2^n entrées et une sortie qui peut être l'image de l'une des 2^n entrées en fonction de l'adresse n appliquée.

Exemples

4 entrées, 1 sortie 2 fils d'adresses (fig. 5.1)

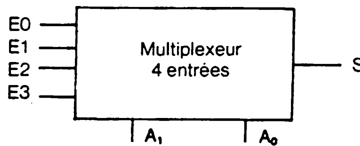


Fig. 5-11. Synoptique d'un multiplexeur 4 entrées.

Table de vérité

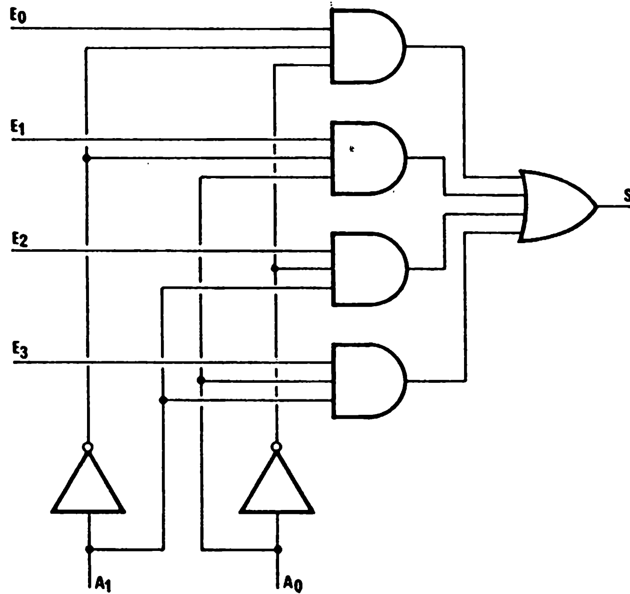
Entrées d'adresses		Sorties d'information
A_1	A_0	S
0	0	E_0
0	1	E_1
1	0	E_2
1	1	E_3

Exercice

– Tirer l'équation de S à partir de la table de vérité.
– Faire le schéma du circuit.

Solution :

$$S = E_0 \cdot \overline{A}_0 \cdot \overline{A}_1 + E_1 \cdot A_0 \cdot \overline{A}_1 + E_2 \cdot \overline{A}_0 \cdot A_1 + E_3 \cdot A_0 \cdot A_1$$



Exemples de multiplexeurs

Type 4519 : 4 fois 2 entrées 1 sortie (fig. 5.12).

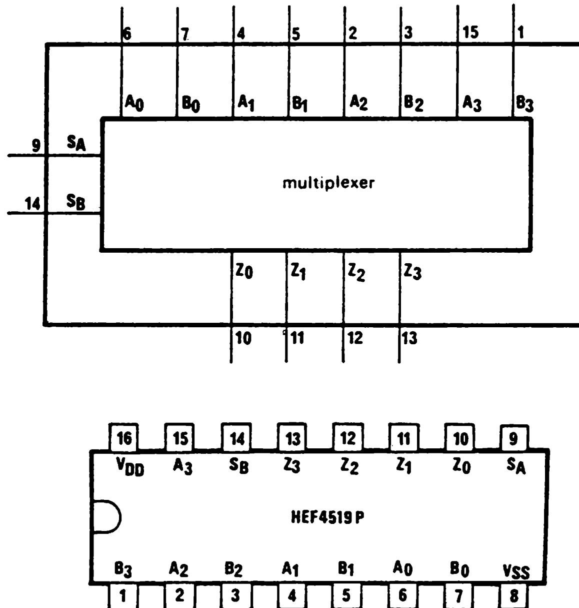


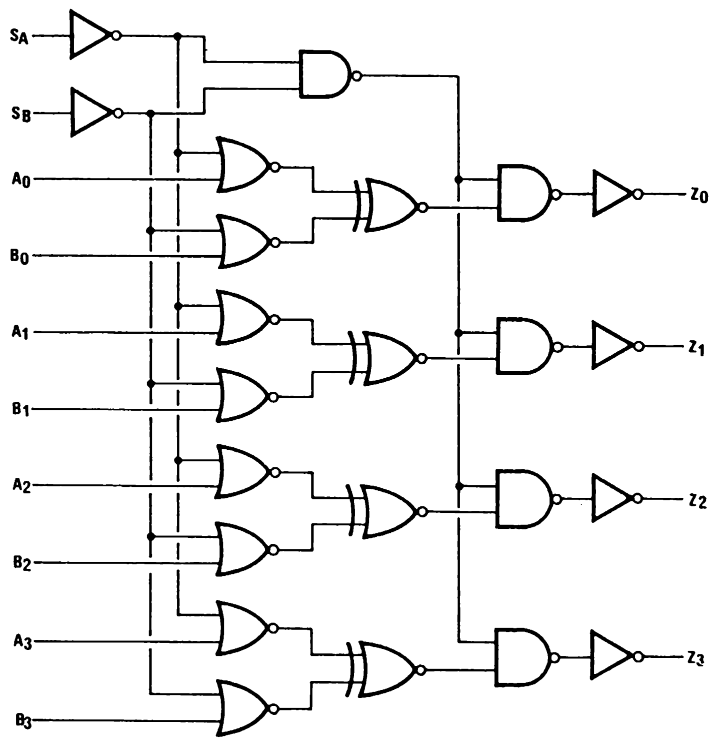
Fig. 5-12. Exemple de circuit multiplexeur.

Remarque :

Ce circuit peut servir à aiguiller une information de 4 bits.

1er mot : A_3 A_2 A_1 A_0
2e mot : B_3 B_2 B_1 B_0

Schéma logique et table de vérité du 4519 (fig. 5.13)



Inputs				Output
S_A	S_B	A_n	B_n	Z_n
L	L	X	X	L
H	L	L	X	L
H	L	H	X	H
L	H	X	L	L
L	H	X	H	H
H	H	L	L	H
H	H	L	H	L
H	H	H	L	L
H	H	H	H	H

Fig. 5-13

Applications des multiplexeurs :

- Réaliser des aiguillages d'informations.
- Lire l'état d'une entrée en fonction de l'adresse n de l'entrée (1 parmi 2^n).
- Permettre la « sérialisation » d'une information, c'est-à-dire la transformation parallèle-série.

Exemple

Principe transformation parallèle-série (fig. 5.14)

Le compteur, à chaque impulsion d'horloge, délivre sur les entrées d'adresse A et B du multiplexeur les informations (00, 01, 10, 11, 00...) ce qui implique sur la sortie, successivement les informations E_0, E_1, E_2, E_3, E_0...

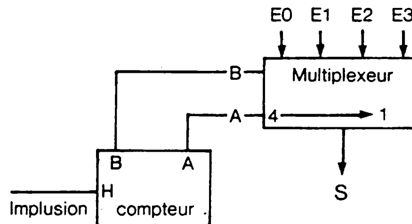
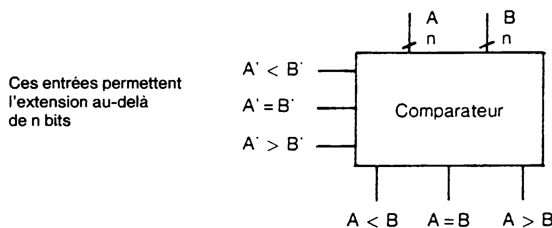


Fig. 5-14. Transformation parallèle-série.

• Comparateur binaire :

Définition

C'est un circuit qui compare 2 mots de n bits chacun en indiquant sur ses sorties si le premier mot est plus grand que le second, si le premier mot est plus petit que le second, si les deux mots sont égaux (fig. 5.15).



Exemples de circuits

Le TTL 7485 (fig. 5.16)

5485 (J, W)
54LS85 (J, W)
7485 (N)
74LS85 (N)

DATA INPUTS
V_{CC} A3 B2 A2 A1 B1 A0 B0
16 15 14 13 12 11 10 9
1 2 3 4 5 6 7 8
B3 DATA INPUT; A<B A=B A>B CASCADING INPUTS; A>B A=B A<B OUTPUTS; GND

Entrées de comparaison (Comparing inputs)				Entrées de cascade (Cascading inputs)			(Sorties Outputs)		
A 3, B 3	**A2, B2**	**A1, B1**	A0, B0	**A > B**	**A < B**	**A = B**	**A > B**	**A < B**	**A = B**
A3 > B3	X	X	X	X	X	X	H	L	L
A3 < B3	X	X	X	X	X	X	L	H	L
A3 = B3	A2 > B2	X	X	X	X	X	H	L	L
A3 = B3	A2 < B2	X	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 > B1	X	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 < B1	X	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 > B0	X	X	X	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 < B0	X	X	X	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	L	L	H	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	H	L	L	H	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	H	L	L	H

85, LS 85

A3 = B3	A2 = B2	A1 = B1	A0 = B0	X	X	H	L	L	H
A3 = B3	A2 = B2	A1 = B1	A0 = B0	H	H	L	L	L	L
A3 = B3	A2 = B2	A1 = B1	A0 = B0	L	L	L	H	H	L

Fig. 5-16. Brochage et table de vérité du 7485.

Schéma logique du 7485 (fig. 5.17)

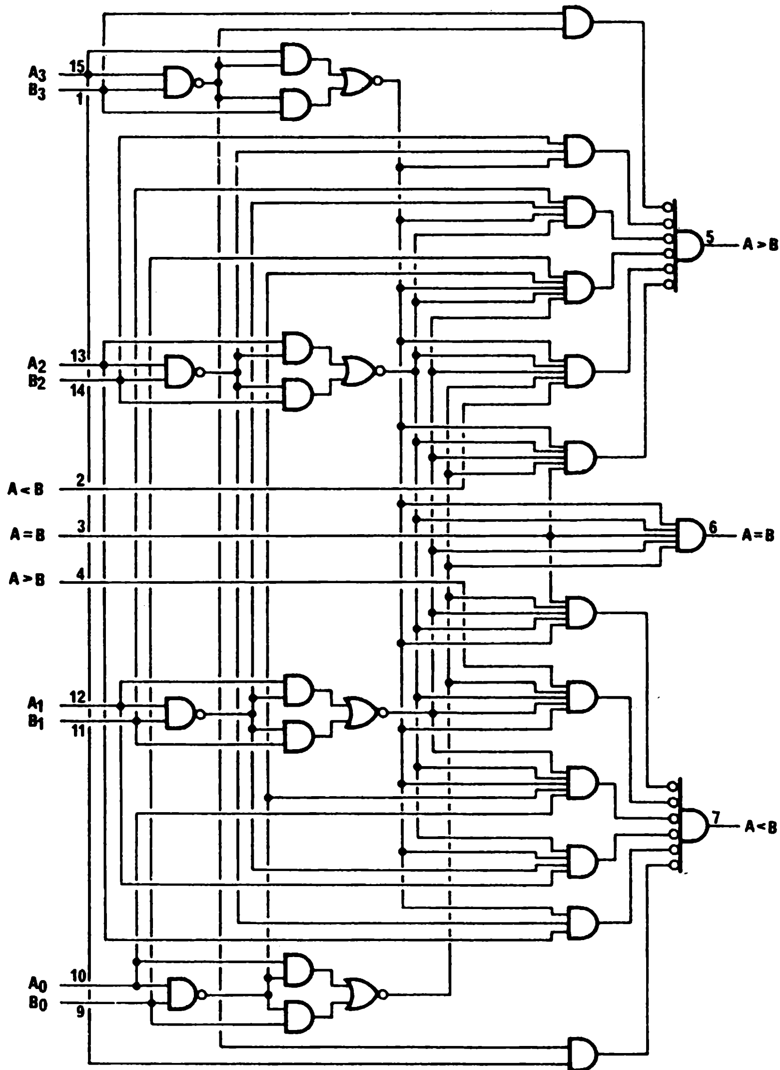


Fig. 5-17

Exemple d'utilisation de ces boîtiers pour la comparaison de 2 mots de n bits (fig. 5.18).

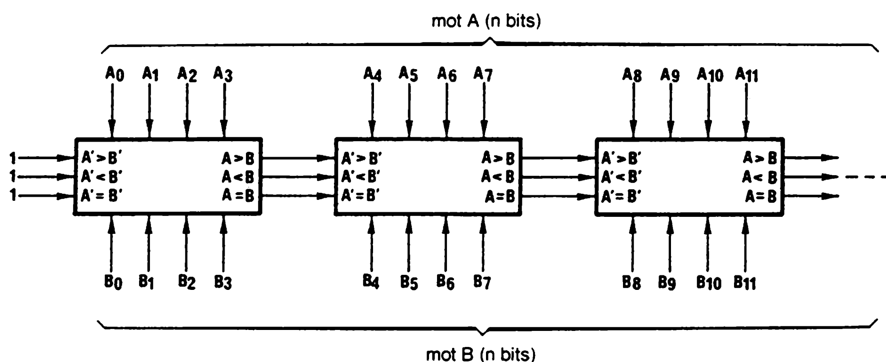


Fig. 5-18. Branchement en cascade de plusieurs boîtiers.

Remarque :

Les comparaisons ainsi réalisées (en cascade) ajoutent les temps de réponse de chaque module.

Exemple d'application :

Soit à reconnaître une adresse parmi 16, dans le cas d'un décodage du bus d'adresses d'un microsystème, vers une sortie L.

La sortie L sera active quand $A_0 = A_1 = A_2 = A_3 = 1$ (c'est-à-dire pour l'adresse $A_3\ A_2\ A_1\ A_0 = 1111$).
C'est cette adresse qui a été sélectionnée, parmi 16, par les interrupteurs et « reconnue » par le comparateur quand il y aura identité entre A et B.

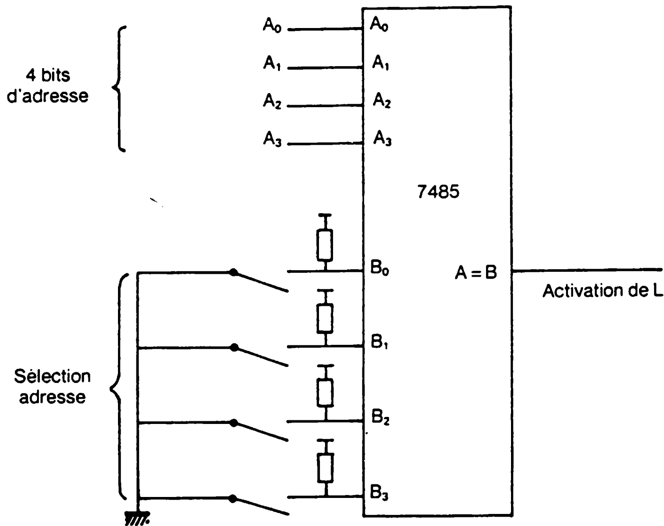


Fig. 5-19. Utilisation d'un comparateur pour le décodage d'adresses.

VI - Additionneurs

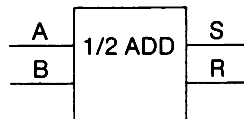
1. Demi-additionneurs

C'est un circuit qui réalise l'addition de deux éléments binaires.

Table de vérité

entrées		sorties		
A	B	R	S	« Plus » S = A + B
0	0	0	0	R = retenue
0	1	0	1	
1	0	0	1	
1	1	1	0	

Schéma (fig. 6.1)



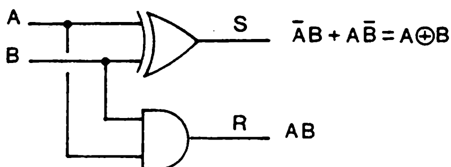
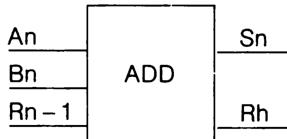


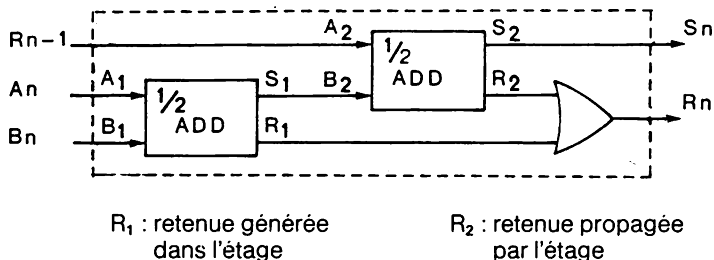
Fig. 6-1. Demi-additionneur.

2. Etage additionneur

Pour réaliser l'addition de 2 nombres binaires, il faut tenir compte lors de l'addition de 2 bits de rang (n) de la retenue de rang (n – 1).
L'opérateur qui réalise l'addition de 2 bits en tenant compte d'une éventuelle retenue en entrée s'appelle étage additionneur (fig. 6.2).



Schéma



R_1 : retenue générée dans l'étage

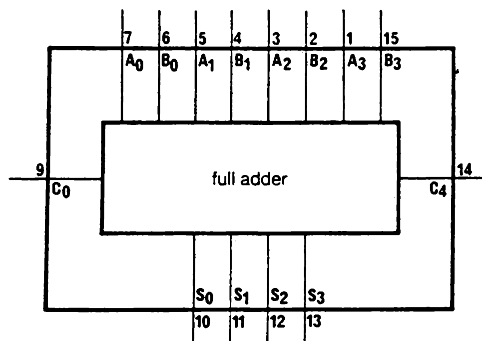
R_2 : retenue propagée par l'étage

Fig. 6-2. Additionneur.

3. Exemples de circuits additionneurs

TTL : 74 LS 83 (additionneur 4 bits)

CMOS : 4008 (additionneur 4 bits) (fig. 6.3)



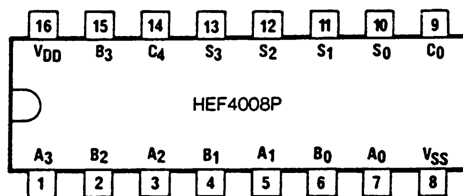


Schéma logique (Logic Diagram)

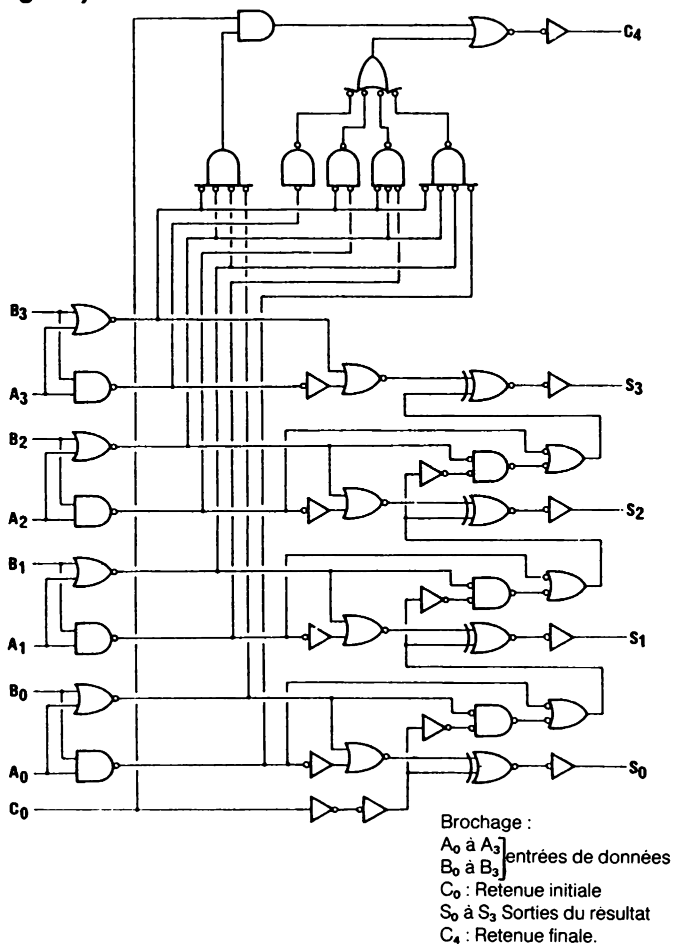


Fig. 6-3. Additionneur 4008.

4. Mise en cascade d'additionneurs

(addition de 2 mots de n bits) (fig. 6.4)

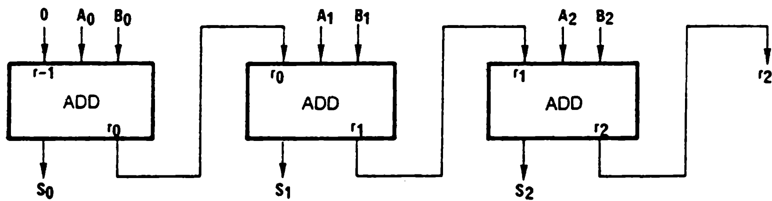


Fig. 6-4. Branchements de plusieurs additionneurs.

5. Soustracteurs

• Demi-soustracteur : D = A moins B

Table de vérité

A	B	R	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

Equations

$D = \bar{A}B + A\bar{B} = A \oplus B$

$R = \bar{A}B$

Schéma (fig. 6.5)

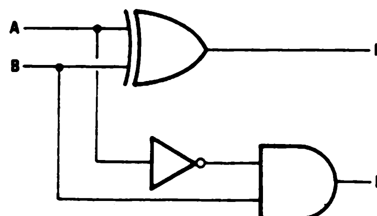


Fig. 6-5. Demi-soustracteur.

• Soustracteur :

Par analogie avec ce que nous avons fait pour l'étage additionneur, l'étage de soustracteur peut s'obtenir en retranchant successivement Bn de An, puis Rn – 1 de la différence D1 obtenue (fig. 6.6)

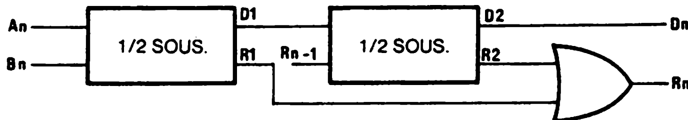


Fig. 6-6. Schéma équivalent du soustracteur.

6. Additionneurs/soustracteurs

• Principe :

Si nous travaillons avec des nombres binaires à n bits, on peut représenter les nombres compris entre 0 et $2^n - 1$.

Exemple : avec 8 bits on peut représenter tous les nombres compris entre 0 et 255 ($2^8 - 1$).
Les étages additionneurs, qui traitent des nombres à n bits, comportent un nombre fini (n) de cellules, c'est-à-dire que le résultat ne peut être donné qu'avec n bits significatifs.
On peut donc écrire : $A + B = A + B \pm 2^n$ (2^n dépasse la taille de l'additionneur à n bits).

Pour réaliser la soustraction de A avec B, on peut écrire aussi :
$A - B = A - B \pm 2^n$.
Si on veut utiliser un additionneur en soustracteur, il faudrait pouvoir additionner à A le nombre (–B).
Trouvons un équivalent de (–B).
Si on fait l'addition de 2 nombres complémentaires, on vérifie que le résultat est égal à $2^n - 1$.

Exemple Si $B = 1001\ 0011$

$\bar{B} = 0110\ 1100$

$B + \bar{B} = 1111\ 1111 = 2^n - 1$.

Si $B + \bar{B} = 2^n - 1$
On peut écrire $B = 2^n - 1 - \bar{B}$ ou encore $-B = \bar{B} + 1 - 2^n$

A − B devient :

$A - B = A - 2^n + 1 + \overline{B}$.

Et comme vu précédemment 2^n dépasse la taille de l'additionneur à n bits, on peut écrire :

$A - B = A + \overline{B} + 1$

Résumé :

Pour faire la soustraction de 2 nombres, on additionne le premier au complémentaire du deuxième (complément à 1) et on ajoute 1.

Remarque :

On peut écrire que $-B = \overline{B} + 1$.

C'est ce qu'on appelle le complément à 2.

• Schéma de principe : Additionneur-Soustracteur (fig. 6.7)

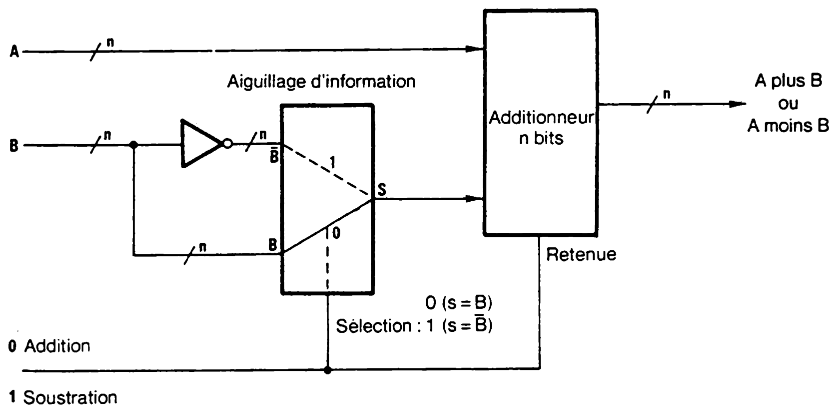


Fig. 6-7. Principe d'un additionneur-soustracteur (n bits).

7. Représentation des nombres signés

On a vu précédemment que l'on pouvait exprimer un nombre négatif par le complément logique à 2 de ce nombre.

Représentons tous les nombres (positifs et négatifs) à l'aide de mots de 4 bits (2^4 combinaisons).

Et vérifions la règle suivante :

Avec n bits, on peut traduire les nombres compris entre

-2^{n-1} et $2^{n-1}-1$.

+7	0111
+6	0110
+5	0101
+4	0100
+3	0111
+2	0010
+1	0001
0	0000
−1	1111
−2	1110
−3	1101
−4	1100
−5	1011
−6	1010
−7	1001
−8	1000

On remarque que le bit de poids le plus fort (ici le bit 3) est significatif du signe :

« 0 » nombres positifs

« 1 » nombres négatifs

VII - Bascules

1. Préliminaire

Nous commençons avec les bascules, l'étude des systèmes séquentiels asynchrones et synchrones qui vont nous permettre, associés aux systèmes combinatoires, d'élaborer des ensembles logiques complets.

2. Introduction sur les bascules

Ces bascules ont pour rôle de mémoriser une information élémentaire. Ce sont donc des **mémoires** élémentaires.

3. Généralités

Une bascule possède deux sorties complémentées Q et $\bar{Q}$. Selon que la sortie Q est à 1 (niveau haut de la tension) ou à 0 (niveau bas de la tension), on dit que la bascule est à 1 ou 0 ; on a stocké l'information 1 ou 0.
On distingue principalement cinq types de bascules.

4. Bascule RS

C'est la bascule élémentaire. Elle peut être réalisée avec des portes NOR (fig. 7.1).

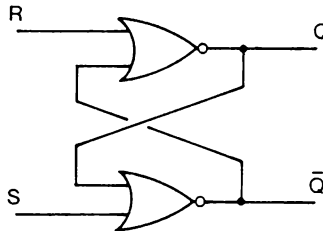


Fig. 7-1. Bascule RS.

• Fonctionnement :

R	S	Qn + 1
0	0	Qn
0	1	1
1	0	0
1	1	indéterminé

Qn = Etat de la sortie avant d'appliquer les niveaux 0 ou 1 sur les entrées R, S.
Qn + 1 = Etat de la sortie après avoir appliqué les niveaux 0 à 1 sur les entrées R, S.

• Conclusion :

Si on applique : R = 1, S = 0   Q = 0
Si R revient à 0, Q reste à 0
R est l'entrée de mise à 0 de la bascule (RESET)
R = 0, S = 1   Q = 1
Si S revient à 0, Q reste à 1
S est l'entrée de mise à 1 de la bascule (SET)
Enfin, si R = S = 1, on impose aux deux sorties l'état 0.
Ce cas est inutilisable.

• Représentation : (fig. 7.2)

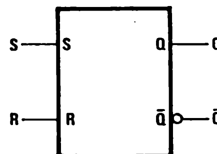


Fig. 7-2. Représentation bascule RS.

5. Bascule $\overline{R}\overline{S}$

On peut la construire à partir des opérateurs NAND (fig. 7.3).

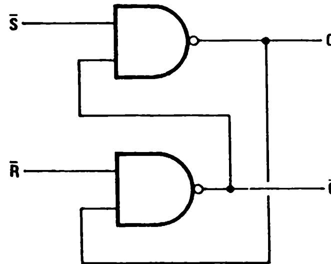


Fig. 7-3.

• Fonctionnement

$\overline{S}$	$\overline{R}$	Q_{n+1}
1	1	Q
1	0	0
0	1	1
0	0	Indéterminé

Remarque :

Les variables d'entrée sont appelées $\overline{R}$ et $\overline{S}$ et cela nous donne une table de vérité identique à celle de la bascule R.S.

• Exemple d'application

Le dispositif anti-rebond

Problème

Si nous désirons commander un système logique à l'aide de bouton-poussoir ou d'inverseurs, un défaut important apparaît : le rebond (fig. 7.4).

Il apparaît à la sortie du circuit logique une succession de créneaux. Ce signal peut être interprété comme une succession d'appuis sur le bouton-poussoir alors que l'opérateur n'a effectué qu'une action. Pour supprimer ce défaut, on peut réaliser le montage suivant : (fig. 7.5).

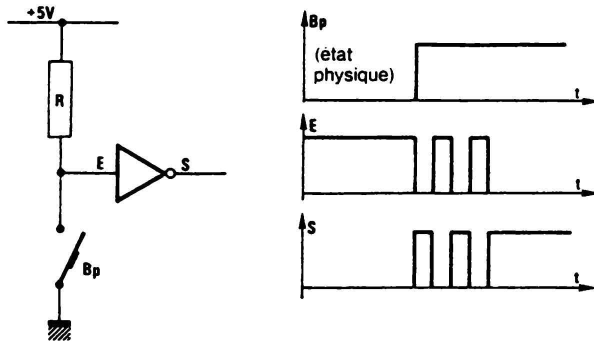


Fig. 7-4. A la fermeture le rebond du bouton-poussoir Bp provoque une suite de créneaux indésirables.

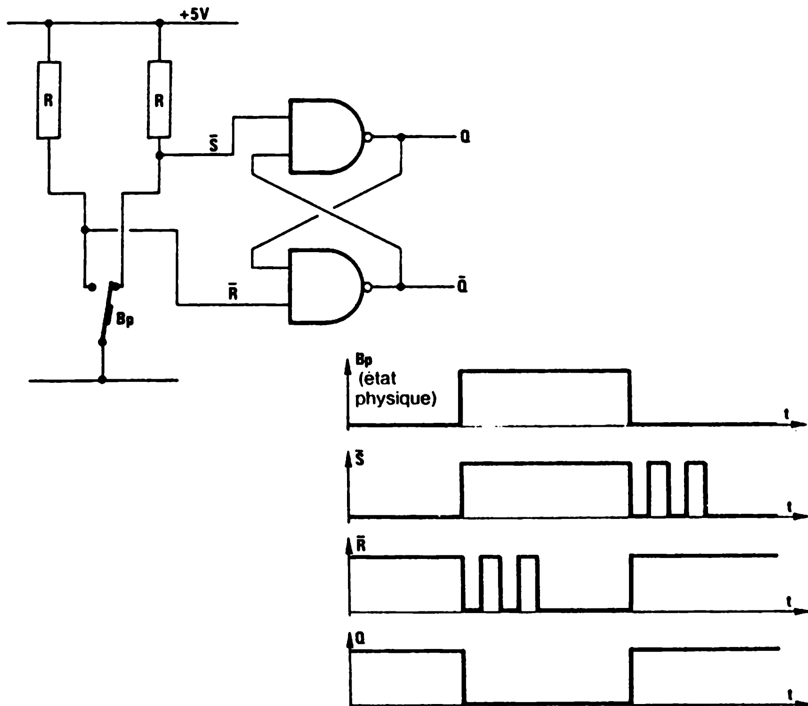


Fig. 7-5. Montage anti-rebond réalisé à l'aide de la bascule $\overline{R}\overline{S}$.

Conclusion :

La bascule $\overline{R}\overline{S}$ élimine le phénomène de rebonds mais l'utilisation d'un bouton-poussoir avec contact travail/repos est nécessaire.

6. Bascule R.S.T.

• Principe

Les conditions d'entrée sont toujours appliquées aux deux entrées R et S mais elles ne sont prises en compte qu'au moment du passage de 0 à 1 (ou 1 à 0 suivant le type de circuit) d'un signal d'horloge T.
Nous réaliserons donc avec la bascule R.S.T. des systèmes séquentiels synchrones (fig. 7.6).

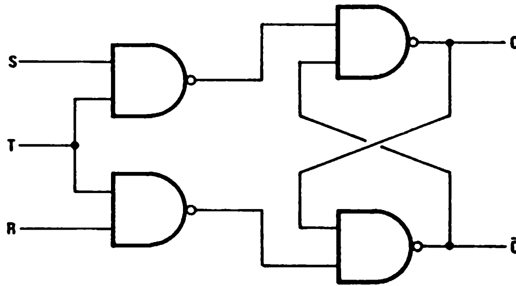


Fig. 7-6. Bascule RST.

• Fonctionnement

La colonne t_{n+1} précise l'état de Q juste après le front de commande de T en fonction des états de R et S avant l'apparition de ce front (colonnes tn).

tn		t_{n+1}
R	S	Q_{n+1}
0	0	Qn
0	1	1
1	0	0
1	1	indéterminé

• Conclusions :

Le basculement éventuel est provoqué par l'entrée T et ne dépend pas du moment où l'on applique les niveaux R ou S.
C'est une bascule RS rendue synchrone par le signal d'horloge T (fig. 7.7). L'entrée T est appelée **entrée dynamique.**

• Représentation :

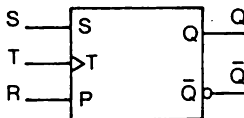


Fig. 7-7. Bascule RST, entrée dynamique ("1"actif).

Remarque
Un rond placé devant le triangle indique que la bascule est sensible au front de descente de l'horloge (fig. 7.8).

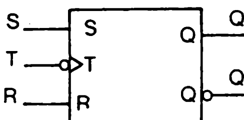


Fig. 7-8. Bascule RST, entrée dynamique ("0" actif).

7. Bascule type D

• Principe :

Cette bascule possède une seule entrée :
L'entrée D (DATA), c'est aussi un circuit de type **synchrone** puisqu'un signal d'horloge est nécessaire pour mémoriser la donnée D.

tn D	tn + 1 Qn + 1
0	0
1	1

On trouve deux types de bascules type D correspondant à deux fonctionnements légèrement différents.

• Bascule Latch : (type D-Latch)

C'est une bascule dérivée de la bascule R.S.T. (fig. 7.9).

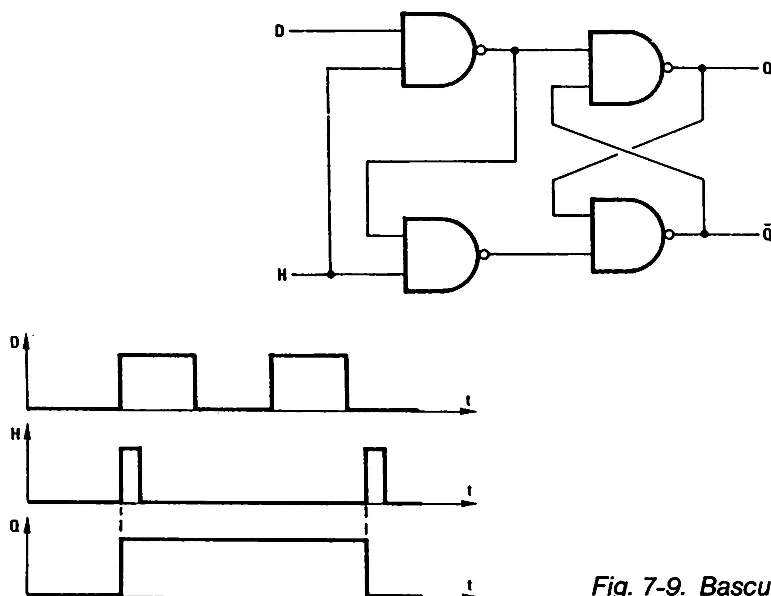


Fig. 7-9. Bascule D.

Lorsque l'entrée d'horloge H est au niveau « 1 » la bascule transfère sur Q l'information présente sur D (Q = D).
Lorsque le signal d'horloge repasse au niveau « 0 » ; la bascule étant «isolée», la dernière information reçue est mémorisée.

Remarque :

La sortie Q prend la valeur de l'entrée D quand le signal d'horloge est à 1. Il ne peut y avoir mémorisation de D que lorsque H est repassé par 0. On peut quand même dire que cette bascule est synchronisée sur le front montant de H.

• Bascule D à commande sur front : (D type edge triggered flip-flop)

Dans ce deuxième type de bascule D, la synchronisation a également lieu sur un front et c'est la valeur de D présente au moment de la **transition de H** qui est introduite dans la bascule (front montant ou descendant).

Exemples de circuits :

- Bascule CMOS 4013 avec 2 entrées de forçage qui permettent de mettre la bascule à 1 (mettre 1 sur S) ou à 0 (mettre 1 sur C).
- Bascule TTL 7474.

Schéma logique de la bascule D (TTL 7474) (fig. 7.10)

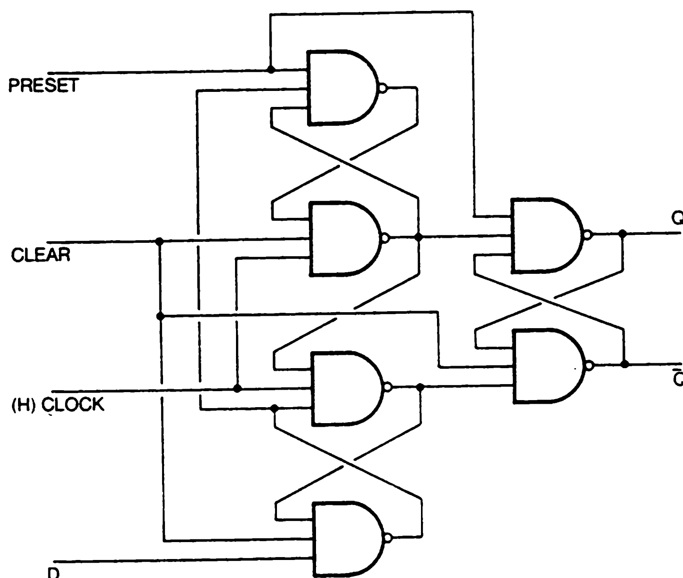


Fig. 7-10. Bascule D type 7474.

En fonctionnement normal, les entrées CLEAR et PRESET doivent être à 1 ; ces entrées de forçage sont dites asynchrones car elles mettent la sortie à 1 ou 0 indépendamment de l'horloge.
Ces entrées permettent de placer un ensemble de bascules dans un état logique donné, à la mise sous tension par exemple pour éviter tout aléa de fonctionnement.

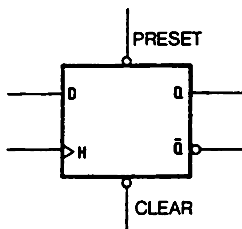


Fig. 7-11.

8. Bascule maître-esclave (master-slave Flip-Flop)

Elle est composée de deux étages : le maître qui reçoit les entrées et l'esclave qui délivre les sorties complémentées Q et $\bar{Q}$. L'étage maître placé en amont commande l'étage esclave.

• Fonctionnement :

La mémorisation de l'information élémentaire se fait en 2 temps :

- **1er temps :** le signal d'horloge passe du niveau 0 au niveau 1.
  Le maître réagit suivant l'état des entrées, l'esclave est **isolé.**
- **2e temps :** le signal d'horloge passe du niveau 1 au niveau 0.
  Le maître est isolé des entrées et transmet son état logique à l'esclave.

L'avantage est de stocker l'information dans un premier temps, d'isoler les entrées, et de la transmettre dans un deuxième temps sur les sorties.

• Bascule R.S. maître-esclave :

La bascule RS maître-esclave est l'association de deux bascules RST. La première bascule reçoit les entrées R et S, c'est le « maître », la seconde est l'esclave (sous les ordres du maître). Le signal d'horloge est inversé sur l'esclave (fig. 7.12).

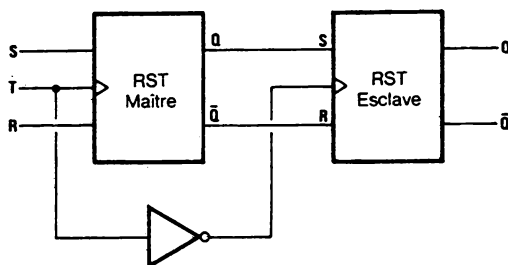


Fig. 7-12. Schéma équivalent de la bascule RS maître-esclave.

La table de vérité reste toujours celle de la bascule RST (indétermination si R = S = 1).

Représentation : (fig. 7.13)

Les entrées asynchrones PRESET et CLEAR sont actives au niveau 0.

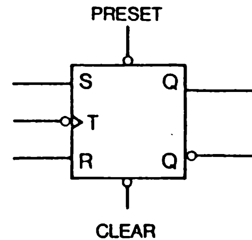


Fig. 7-13. Bascule RS maître-esclave.

• Bascule JK maître-esclave

On lève l'indétermination de la bascule RST (R = S = 1), en complémentant R et S, en utilisant les sorties de la bascule. Les nouvelles entrées s'appellent J et K. Les entrées S et R de la bascule RS « maître » peuvent s'écrire :

$S = J\bar{Q}$ et $R = K\,Q$ (fig. 7.14)

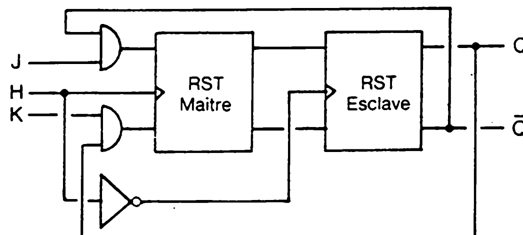


Fig. 7-14. Schéma équivalent de la bascule JK maître-esclave.

On peut vérifier que dans le cas de J = K = 1, la sortie Q change d'état à chaque impulsion d'horloge.

Table de vérité de la bascule JK

tn		tn+1
J	K	Q_{n+1}
0	0	Qn
0	1	0
1	0	1
1	1	$\bar{Q}n$

Entrées asynchrones :
PRESET = 0 → Q = 1
CLEAR = 0 → Q = 0

Représentation : (exemple)

Rappel. Les entrées asynchrones doivent être à 1 pour assurer un fonctionnement correct en synchronisme avec H selon la table de vérité (fig. 7.15).

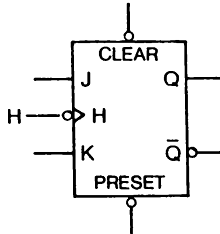


Fig. 7-15. Bascule JK.

• Bascule JK maître-esclave à verrouillage :

Une caractéristique principale de la bascule précédente est qu'elle est sensible au **niveau** de H.
C'est-à-dire que pendant que H est au niveau 1, si les entrées évoluent, les sorties feront de même. Dans certaines applications il est intéressant d'avoir une bascule qui soit sensible au **front** du signal d'horloge H. C'est ce qui se passe dans la bascule JK maître-esclave à verrouillage. Les données ne sont prises en compte que pendant un temps très court (20ns) au moment du passage de 0 à 1 du signal d'horloge.
L'avantage supplémentaire d'un tel fonctionnement est que les éventuels parasites, présents sur les entrées, ont leur influence fortement diminuée puisque seul l'état présent est pris en compte pendant un temps très bref.

9. Bascule JK à déclenchement sur front
(JK edge-triggered Flip-Flop)

Elles sont comme dans le cas précédent, déclenchées lorsque le signal d'horloge passe de 0 à 1 (front montant) mais elles ne sont pas maître-esclave.

Un premier exemple a déjà été rencontré avec la bascule type D edge triggered.

Remarque :

Selon qu'il s'agit du front montant ou descendant de l'horloge, la bascule est dite **positive ou négative edge-triggered.**

Exemple de circuit (CMOS)

– 4027

Représentation : (fig. 7.16)

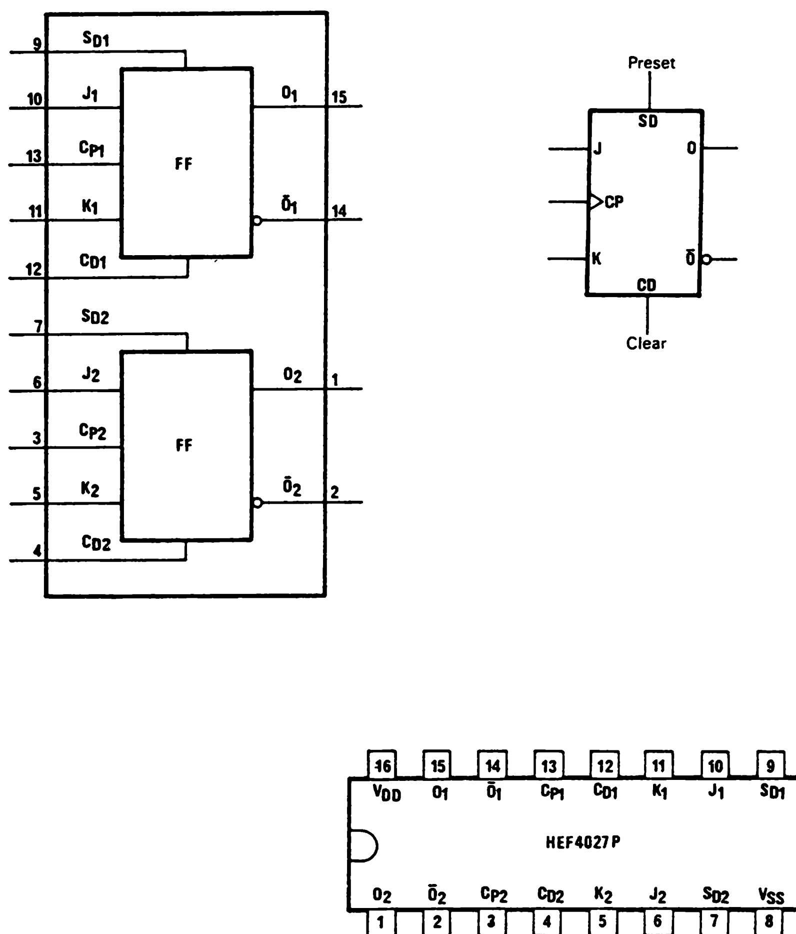


Fig. 7-16. Représentation et brochage de la 4027.

Table de vérité et désignation des broches de la 4027

Inputs (entrées)					Outputs (sortie)	
S_D	C_D	C_P	J	K	Q_{n+1}	$\bar{Q}_{n+1}$
H	L	X	X	X	H	L
L	H	X	X	X	L	H
H	H	X	X	X	H	H
L	L	∫	L	L	no change inchangées	
L	L	∫	H	L	H	L
L	L	∫	L	H	L	H
L	L	∫	H	H	$\bar{Q}_n$	Q_n

PINNING

J.K. : synchronous inputs

CP : clock input (LOW to HIGH edge-triggered)

S_D : asynchronous set-direct input (active HIGH)

C_D : asynchronous clear-direct input (active HIGH)

Q : true output

$\bar{Q}$ complement output

H : HIGH state (the more positive voltage)

L : LOW state (the less positive voltage)

X : state is immaterial

∫ : positive-going transition

Q_{n+1} : state after clock positive transition

BROCHAGE

J.K. : entrées synchrones

CP : entrée d'horloge (déclenchée sur le front montant)

S_D : entrée asynchrone de forçage à 1 (actif au niveau haut)

C_D : entrée asynchrone de forçage à 0 (actif au niveau haut)

Q : sortie de la bascule

$\bar{Q}$: sortie complémentée de la bascule

H : niveau haut

L : niveau bas

X : état indifférent

Q_{n+1} : état de la sortie après le front d'horloge

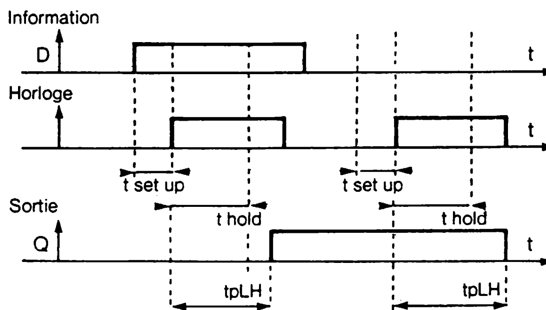
Fig. 7-17. Circuit 4027.

10. Paramètres dynamiques d'une bascule

Le constructeur nous donne un certain nombre de paramètres dynamiques que nous allons définir.
L'utilisateur de la bascule devra en tenir compte pour un fonctionnement correct. Ces paramètres dynamiques sont :

- **Fréquence maximale de l'horloge :** l'utilisateur ne peut demander à la bascule un fonctionnement à une fréquence supérieure à celle définie par le constructeur.
- **Temps de préaffichage** (Set up time) : l'information présente sur les entrées dépendant de l'horloge (JK par exemple) doit être présente un certain temps avant le front actif de l'horloge.
- **Temps de maintien** (Hold time) : l'information présente sur les entrées dépendant de l'horloge doit être maintenue un certain temps après le front actif de l'horloge.
- **Temps de basculement** (tpLH : low to high et tpHL : high to low). C'est le temps que met la sortie pour changer de niveau.

Exemple - Diagrammes bascule D



VIII - Registres

Un registre permet d'emmagasiner une information binaire.
Cette information peut être lue à tout moment. On distingue les registres de mémorisation et à décalage. Ils sont constitués tous les deux de bascules RS, D ou JK.

1. Registres de mémorisation

• Principe :

Il permet la mémorisation d'un mot de n bits. Il est constitué de n cellules indépendantes. Chacune mémorisant 1 bit du mot.

• Réalisation avec des bascules RS : (fig. 8.1)

Deux fils de commande sont nécessaires, le premier pour la remise à zéro (RAZ) du registre, le deuxième pour la mémorisation (M).

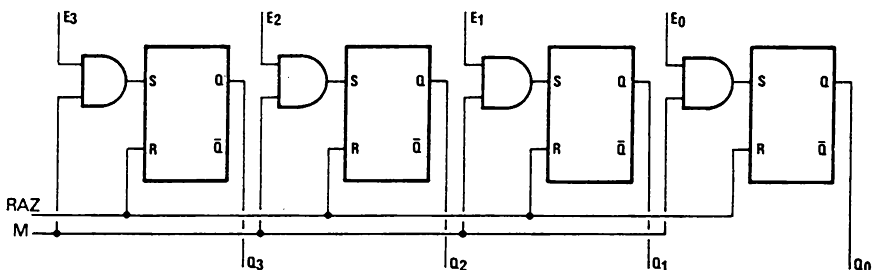
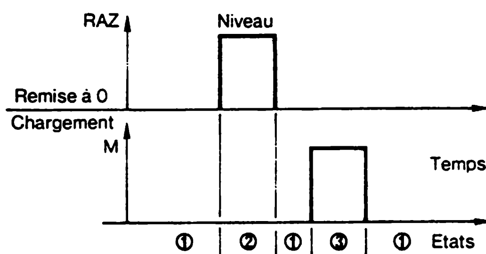


Fig. 8-4. Registre 4 bits réalisé avec des bascules RS.

Table de vérité

RAZ		M	Sorties (Q_3 Q_2 Q_1 Q_0)
1	0	0	Le registre conserve son information.
2	1	0	Remise à 0 ($Q_3 = Q_2 = Q_1 = Q_0 = 0$)
3	0	1	Le registre mémorise l'information présente sur les entrées E_0, E_1, E_2, E_3 (Q_i reçoit E_i).

Fonctionnement :

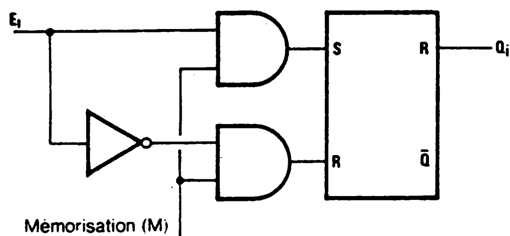


Inconvénient

On voit qu'il est obligatoire de faire une remise à 0 avant de donner l'ordre de chargement.

Amélioration

On peut modifier le montage précédent pour qu'un seul ordre de chargement soit nécessaire.
On obtient pour une cellule le schéma suivant (fig. 8.2).



Fonctionnement : sur ordre de mémorisation (M = 1) $S = E_i$ et $R = \bar{E}_i$ donc $Q_i = E_i$

Fig. 8-2. Cellule RS d'un registre de mémorisation.

• Réalisation avec des bascules D

Actuellement on ne trouve plus de registres intégrés avec des bascules RS. Ils le sont avec des bascules D (fig. 8.3).

Schéma de principe

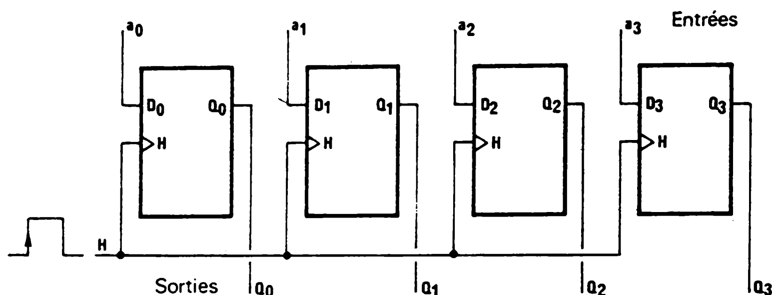


Fig. 8-3. Registre D.

Fonctionnement :
En synchronisme avec le signal d'horloge H, les bascules enregistrent les informations présentes sur les entrées D_0 -D_1 -D_2 - D_3.

• Réalisation avec des bascules JK (fig. 8.4)

Schéma de principe

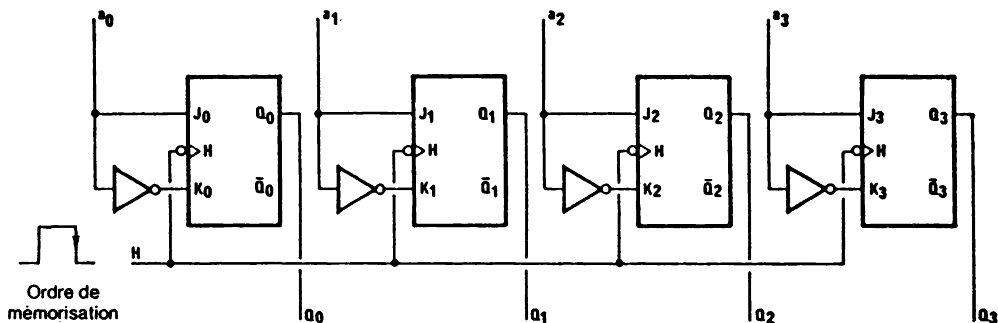


Fig. 8-4. Registre JK.

Fonctionnement :

En synchronisme avec le signal d'horloge H (ici front descendant), le registre mémorise les états des entrées E_0 - E_1 - E_2 - E_3.

En effet, on voit que pour chaque bascule :

$J_i = E_i$ et $K_i = \bar{E}_i$ donc $Q_i = E_i$ (après l'impulsion d'horloge).

2. Registres à décalage

• Principe

Si on désire mémoriser un mot de n bits le registre devra se composer de n cellules identiques (bascules).
Ces bascules seront reliées entre elles de telle manière que le bit de rang p puisse être transmis à la bascule de rang p + 1 sur ordre d'un signal de synchronisation (horloge).

• Mémorisation d'une information

Le mot de n bits peut être introduit dans ce registre soit « en série », soit « en parallèle ».

- **Mode série :** le mot de n bits est appliqué bit à bit sur l'entrée de la première bascule jusqu'à ce que le registre soit plein.
- **Mode parallèle :** le mot est appliqué en une seule fois comme dans les registres précédents.

• Lecture de l'information

La lecture s'effectue sur la sortie dite série, c'est-à-dire la sortie de la dernière bascule (bascule de rang le plus grand) bit à bit, par décalage du mot à chaque impulsion d'horloge.

Remarque

Dans certains registres à décalage, on peut trouver une sortie parallèle permettant la lecture du mot en une seule fois, comme dans les registres précédents.

Si le registre possède plusieurs modes de fonctionnement (entrée série - entrée parallèle - sortie série - sortie parallèle) l'utilisateur pourra sélectionner le mode en fonction de l'application.

Certains de ces registres peuvent aussi décaler à droite et à gauche.

• Exemple de registre à décalage procurant tous les modes de fonctionnement (fig. 8.5)

– 40194 CMOS, 7495 A TTL

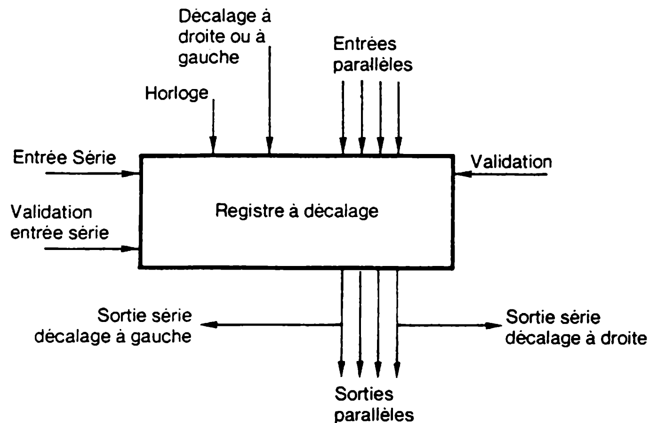


Fig. 8-5. Registre à décalage universel.

• Réalisation

Quel que soit le type de bascule utilisé, le principe est le même :

– les entrées horloge de chaque bascule sont reliées entre elles pour former l'entrée horloge du registre,

– les sorties d'une bascule sont connectées aux entrées de la suivante.

A titre d'exemple, nous allons montrer les schémas de principe des registres à décalage réalisés à partir de bascules D et de bascules JK.

• Registre à décalage entrée série, sortie série ou parallèle (bascule D) (fig. 8.6)

Fonctionnement :

La bascule D est commandée sur le front positif (par exemple) de l'impulsion d'horloge.
La sortie Q est appliquée à l'entrée D de la bascule suivante.
Au niveau de chaque bascule, la sortie Q suit l'entrée D avec un retard τ dû au circuit par rapport au front de montée du signal d'horloge.

Donc la sortie de la nième bascule sera, elle aussi, en retard du temps τ.

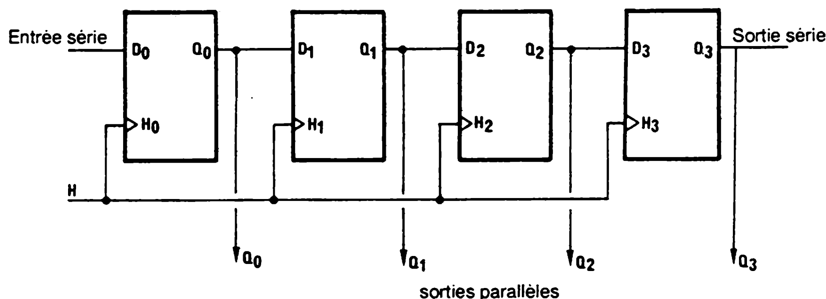


Fig. 8-6. Registre à décalage D.

Exemple de fonctionnement

Avant le 1er front montant de H	Après le 1er front montant de H
$D = 1$	$Q_0 = 1$
$Q_0 = 0$	$Q_1 = 0$
$Q_1 = 1$	$Q_2 = 1$
$Q_0 = 0$	$Q_3 = 0$
$Q_3 = 1$	

- **Registre à décalage série, sortie série ou parallèle (bascule JK)** (fig. 8.7)

Schéma de principe

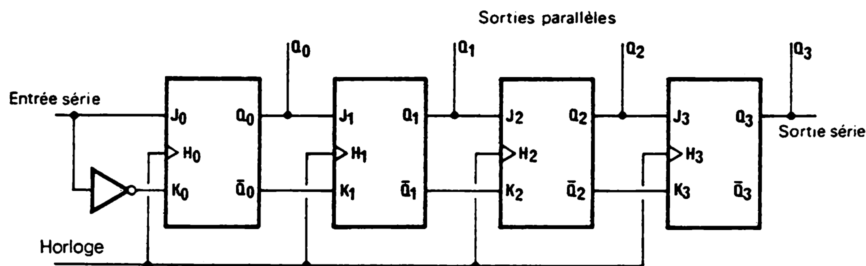


Fig. 8-7. Registre à décalage JK.

Fonctionnement :

On voit que : $J_i = Q_{i-1}$; $K_i = \bar{Q}_{i-1}$; donc à chaque impulsion d'horloge les sorties Q_i et $\bar{Q}_i$ mémorisent les états de J_i et K_i (complémentaires) donc de Q_{i-1} et $\bar{Q}_{i-1}$. Il y a donc décalage.
On remarquera que la première bascule (rang 0) reçoit l'information sur J_0 et K_0 en reçoit le complément.
En effet, si l'entrée vaut « 0 », J = 0 et K = 1 (mise à zéro de la bascule) et si elle vaut J = 1 et K = 0 (mise à 1 de la bascule).

• Exemple de registre à décalage avec chargement série ou parallèle, sortie série ou parallèle (fig. 8.8)

Schéma de principe

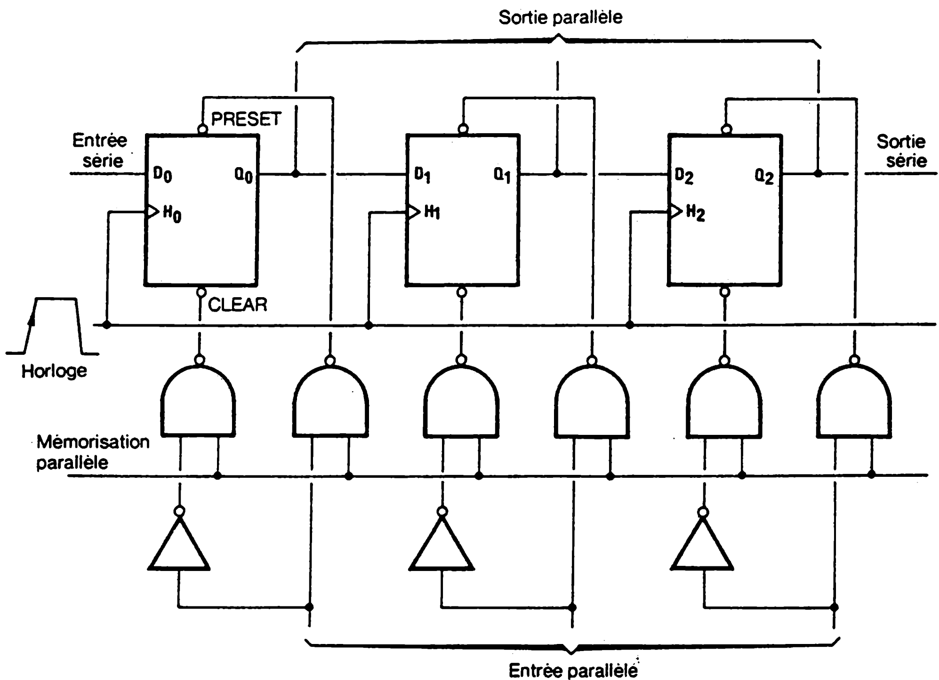


Fig. 8-8. Registre à décalage à entrée série ou parallèle.

Fonctionnement :

Si on applique 1 sur l'entrée **mémorisation parallèle** on agit sur les entrées « CLEAR » et « PRESET » de chaque bascule de rang p.

$CLEAR_p$ = bit (entrée parallèle)

et $PRESET_p = \overline{bit_p}$ (entrée parallèle)

Donc si $bit_p = 0$ CLEAR = 0
PRESET = 1

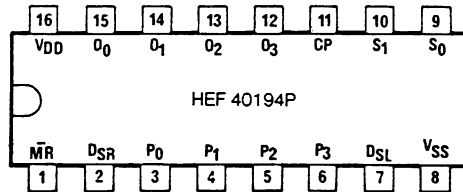
→ mise à 0 de la bascule de rang p

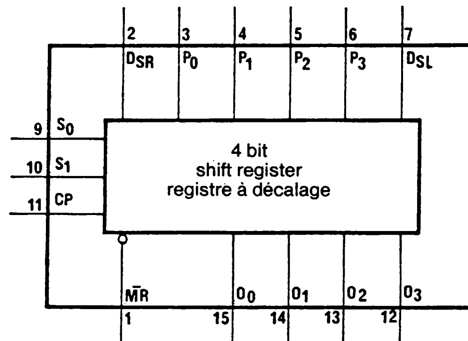
si $bit_p = 1$ CLEAR = 1
PRESET = 0

→ mise à 1 de la bascule de rang p.

Exemple de circuit intégré :

CMOS : 40 194 : 1 registre 4 bits (bascules D) décalage à droite - à gauche (fig. 8.9).





Mode de fonctionnement Operating mode	Entrées ($\overline{MR}$ = HAUT) Inputs ($\overline{MR}$ = HIGH)					Sorties à t_{n+1} Outputs at t_{n+1}			
	S_1	S_0	D_{SR}	D_{SL}	P_0 to P_3	O_0	O_1	O_2	O_3
Maintien **Hold**	L	L	X	X	X	O_0	O_1	O_2	O_3
Décalage à gauche **Shift left**	H H	L L	X X	L H	X X	O_1 O_1	O_2 O_2	O_3 O_3	L H
Décalage à droite **Shift right**	L L	H H	L H	X X	X X	L H	O_0 O_0	O_1 O_1	O_2 O_2
Chargement parallèle **Parallel load**	H H	H H	X X	X X	L H	L H	L H	L H	L H

H = HIGH state (the more positive voltage), état HAUT (tension la plus positive).
L = LOW state (the less positive voltage), état bas (tension la moins positive).
X = state is immaterial, état indifférent.
t_{n+1} = state after next LOW to HIGH transition of CP, état après le front montant suivant de CP.

Fig. 8-9. Représentation et table de vérité du registre 40194.

• Application des registres à décalage

– On remarque qu'un décalage à gauche revient à multiplier le mot enregistré par 2 et à droite à le diviser par 2. C'est une façon simple de multiplier ou de diviser un nombre par 2 dans un calculateur.

– **Sérialisation d'une information :** lorsqu'on veut transmettre sur un fil une information de n bits, il est nécessaire de faire une transformation parallèle-série de l'information c'est-à-dire en effectuant un décalage à droite du mot, en synchronisme avec un signal d'horloge, la ligne étant reliée à la sortie de la bascule la plus à droite.

– **Ligne à retard numérique :** Si on veut créer un retard pour une transmission de données particulières, il suffit de faire passer l'information dans un registre à décalage qui la « freinera » au rythme de l'horloge (fig. 8.10).

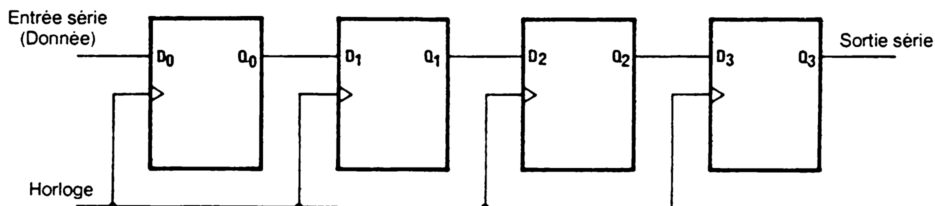


Fig. 8-10. Registre série - série 4 à étages.

Le retard est égal à 4 T, T étant la période de l'horloge (fig. 8.11). Il suffit donc, pour réaliser une ligne à retard numérique, d'utiliser un registre série - série à n bascules pour obtenir un retard de n T, T étant la période du signal de l'horloge.

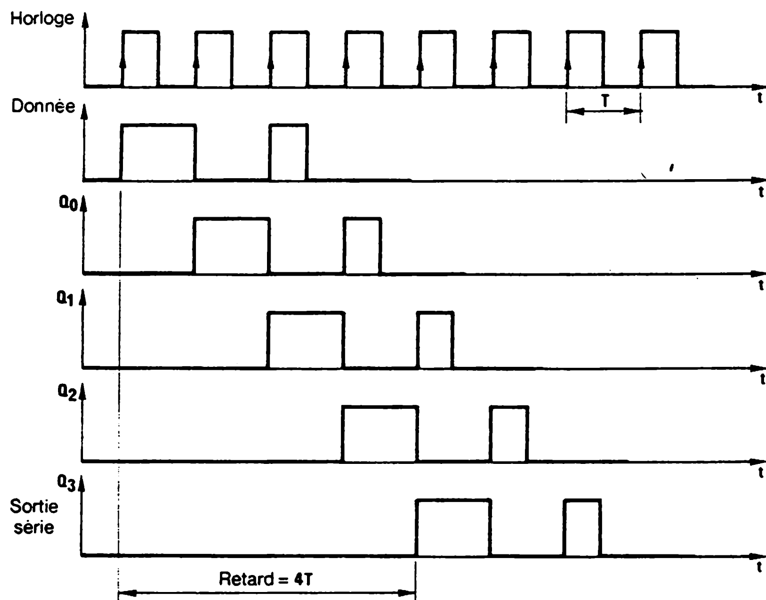


Fig. 8-11. Graphes de signaux.

IX – Compteurs

1. Définition

Ils sont constitués de n bascules, ils peuvent donc mémoriser des mots de n bits (mémoire). Mais ils peuvent aussi décrire, au rythme d'une horloge, une séquence déterminée qui peut avoir au maximum 2^n combinaisons différentes.
Ainsi, un compteur est un système donnant des états stables de ses sorties à chaque impulsion d'horloge. Ce nombre d'états stables est appelé le modulo N de ce compteur.

2. Classement

Les compteurs binaires sont classés en 2 catégories :
- les compteurs asynchrones,
- les compteurs synchrones.

3. Les compteurs asynchrones

- Principe : (avec bascules JK déclenchant sur front descendant) (fig. 9.1)

  Une bascule JK est un diviseur par 2 si les entrées J et K sont au niveau logique 1.

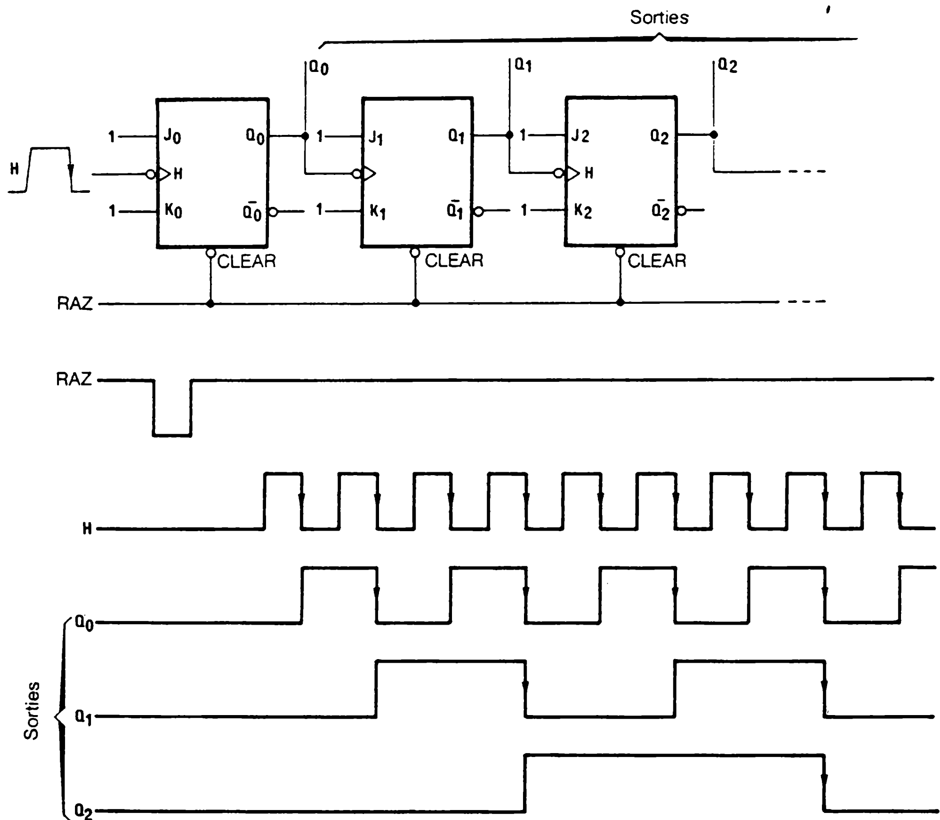


Fig. 9-1. Schéma et diagrammes d'un compteur binaire à bascules JK.

Conclusion :

Le signal d'horloge d'une bascule de rang i est le signal de sortie de la bascule de rang i – 1.

Nota :

Avec des bascules D, pour obtenir un diviseur par 2.
On relie la sortie $\overline{Q}i$ à l'entrée Di de chaque cellule. L'entrée Hi + 1 de la bascule suivante est reliée à la sortie $\overline{Q}i$ de la bascule précédente, dans le cas de bascule D communtant sur le front positif du signal d'horloge (fig. 9.2).

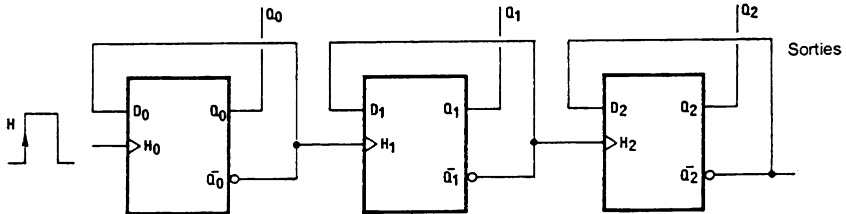


Fig. 9-2. Compteur asynchrone réalisé avec des bascules D.

- **Décompteur asynchrone** : Exemple JK (déclenchant sur front descendant) (fig. 9.3)

Pour décompter, il suffit de relier la sortie $\bar{Q}i$ de la bascule de rang i à l'entrée H_{i+1} de la bascule de rang i + 1.

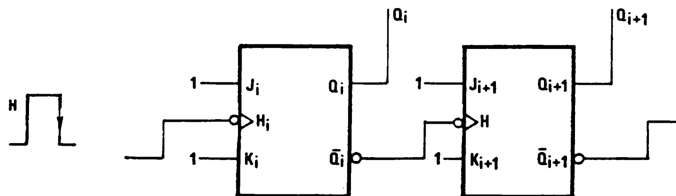


Fig. 9-3. Principe de compteur asynchrone à bascule JK.

- **Compteur décompteur asynchrone** : Exemple JK (déclenchant sur le front descendant).

Il suffit de pouvoir relier Hi + 1 à Qi pour le comptage et Hi + 1 à $\bar{Q}i$ pour le décomptage. D'où l'utilisation d'un multiplexeur à 2 entrées, 1 sortie (fig. 9.4).

On peut donc avec n bascules compter de 0 à $2^n - 1$ ou décompter de $2^n - 1$ à 0. Mais il est intéressant parfois de pouvoir charger le contenu d'un compteur à une certaine valeur comprise entre 0 et $2^n - 1$ et de pouvoir le remettre à 0 après une certaine valeur.
Nous allons donc voir comment charger un compteur à une valeur désirée et comment réaliser un compteur à cycle incomplet.

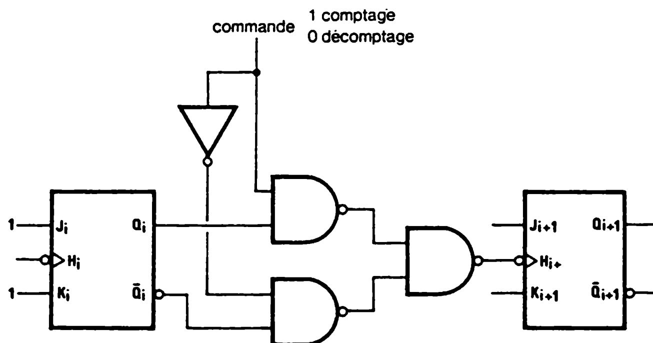
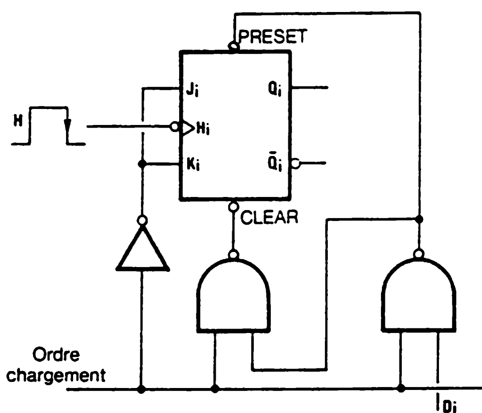


Fig. 9-4. Principe compteur-décompteur asynchrone à bascule JK.

• Principe de chargement d'un compteur à une valeur désirée (fig. 9.5)



Fonctionnement :

A. Entrée « ordre de chargement » à 0
   Ji = Ki = 1
   PRESET = CLEAR = 1 : comptage

B. Entrée « ordre de chargement » à 1
   Ji = Ki = 0 : comptage bloqué

si Di = 0 → PRESET reçoit 1
CLEAR reçoit 0
→ Qi = 0

si Di = 1 → PRESET reçoit 0
CLEAR reçoit 1
→ Qi = 1

Fig. 9-5. Cellule JK d'un compteur.

• **Principe** : compteur binaire à cycle incomplet :

Exemple :

Réaliser une décade de comptage avec 4 bascules JK.

Solution :

On peut par exemple utiliser les entrées asynchrones CLEAR et PRESET pour forcer le compteur à s'arrêter à une certaine valeur pour repartir à 0.

En effet, si on analyse la table de vérité d'un compteur composé de 4 bascules (N max = 2^4 = 16) on voit qu'il suffit de remettre à 0 toutes les bascules pour la combinaison des sorties $Q_3\ Q_2\ Q_1\ Q_0 = 1010_2 = 10_{10}$ pour que le compteur passe à 0000 après 1001, c'est-à-dire passe à 0 après 9.

Si CLEAR est actif au 0, on peut écrire :

$$\text{CLEAR} = \overline{\underset{1}{Q_3}\ \underset{0}{\bar{Q}_2}\ \underset{1}{Q_1}\ \underset{0}{\bar{Q}_0}} = \underset{0}{\bar{Q}_3} + \underset{1}{Q_2} + \underset{0}{\bar{Q}_1} + \underset{1}{Q_0}$$

Remarque

On peut vérifier que le résultat aurait été le même si on avait écrit :
CLEAR $= \bar{Q}_3 + \bar{Q}_1$ (fig. 9.6).

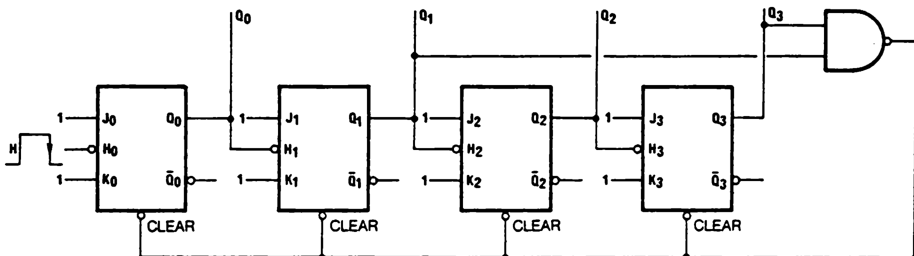


Fig. 9-6. Compteur modulo 10 (décade) asynchrone.

• Inconvénient des compteurs asynchrones

On remarque que l'évolution d'un compteur asynchrone au moment du front de commande de l'horloge présente des aléas (états transitoires non corrects) dus aux temps de réponse des bascules.

Si les sorties du compteur alimentent un système d'acquisition rapide il ne conviendra pas, par contre, si elles sont exploitées par un organe ayant un temps de réponse très grand par rapport à celui du compteur, les états transitoires deviendront « transparents » pour celui-ci.

• Fréquence maximale de fonctionnement

Si on appelle Dp le délai de propagation d'une bascule, la structure d'un compteur asynchrone à N bascules montre que le délai de propagation du bit de poids le plus fort (dernière bascule) se trouve multiplié par le nombre N de bascules ainsi par exemple un compteur modulo 16 (4 bascules) aura un délai de propagation de : $T = Dp \times 4$.

Exemple

Si Dp est de 20 ns, un compteur modulo 16 aura un retard propre de :

$$T = 20\ \text{ns} \times 4 = 80\ \text{ns}$$

On en déduit la fréquence maximale de fonctionnement :

$$f = \frac{1}{2T} = 6{,}25\ \text{MHz}$$

Remarque :

$f = \frac{1}{2T}$ car il faut faire la somme des temps de propagation sur le front de montée et de descente du signal par période.

4. Les compteurs synchrones

• Principe :

Dans un compteur synchrone toutes les bascules composant le compteur reçoivent en même temps le signal d'horloge.

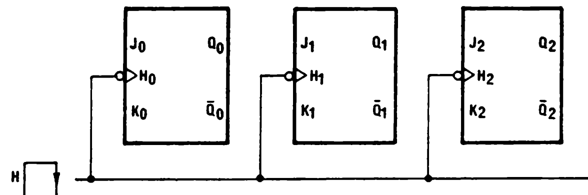


Fig. 9-7. Toutes les entrées d'horloge des bascules JK sont reliées.

Si on veut que le compteur décrive une séquence bien déterminée, il faut « jouer » sur les entrées de commande synchrones (entrées J et K ou D). Les entrées CLEAR et PRESET sont libres, permettant le prépositionnement du compteur (fig. 9.7).

• Fréquence maximale de fonctionnement

Si les temps de retard sont les mêmes pour chaque bascule, il n'y a plus d'états intermédiaires indésirables.
Le délai de propagation est donc égal à celui d'une bascule.

Exemple

Si Dp = 20 ns

$$\text{fréquence maximale} = \frac{1}{2 \times 20 \text{ ns}} = 25 \text{ MHz}$$

Il est donc préférable d'utiliser une structure synchrone pour des applications nécessitant des temps de réponse très grands.

Remarque

Si le compteur synchrone ne présente pas d'états transitoires gênants, sa conception nécessite l'apport d'opérateurs logiques supplémentaires.

• Compteur binaire synchrone : calcul

Les entrées de commande de chaque bascule (J, K ou D suivant le type de bascule) seront fonction des sorties des bascules constituant le compteur.

Nous devons donc étudier l'état initial et l'état final (après le signal d'horloge) de chaque bascule suivant la table de vérité décrivant la séquence à décrire, et nous en déduirons les équations logiques des entrées de commande.

Rappel table de vérité des bascules JK et D :

Fonctionnement désiré		Avant le signal d'horloge les entrées doivent être à :		
Q initial	**Q final**	**J**	**K**	**D**
0	0	0	Ø	0
0	1	1	Ø	1
1	0	Ø	1	0
1	1	Ø	0	1

Ø : état indifférent 0 ou 1.

Pour simplifier les équations, utilisons les tableaux de Karnaugh dans les exemples qui suivent :

• Compteur binaire synchrone à cycle complet :

Exemple avec bascule JK

Etat initial			Etat final			Entrées JK					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	Ø	0	Ø	1	Ø
0	0	1	0	1	0	0	Ø	1	Ø	Ø	1
0	1	0	0	1	1	0	Ø	Ø	0	1	Ø
0	1	1	1	0	0	1	Ø	Ø	1	Ø	1
1	0	0	1	0	1	Ø	0	0	Ø	1	Ø
1	0	1	1	1	0	Ø	0	1	Ø	Ø	1
1	1	0	1	1	1	Ø	0	Ø	0	1	Ø
1	1	1	0	0	0	Ø	1	Ø	1	Ø	1

Du tableau d'analyse on en déduit que les équations des entrées Ji et Ki peuvent s'écrire :

$$J_0 = K_0 = 1$$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0 \,.\, Q_1$$

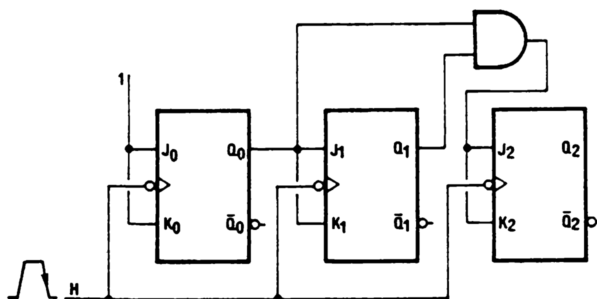


Fig. 9-8. Principe d'un compteur binaire pur synchrone.

Remarques :

– Pour un nombre de cellules supérieur, on peut vérifier que les équations de commutation s'écrivent :

$$J_0 = K_0 = 1$$

$$Ji = Ki = Q_0.Q_1.Q_2...Q_{i-1}$$

D'où le schéma (fig. 9.8)

- Pour obtenir un décômpteur, il suffit de remplacer les Qi par $\bar{Q}i$ dans les équations précédentes comme le compteur binaire asynchrone.

• Compteur binaire synchrone à cycle incomplet :

1er exemple

Compteur modulo 5 avec des bascules JK.
C'est un diviseur par 5, il nécessite 3 bascules (2^3 combinaisons), il faudra donc éliminer 3 combinaisons.

Etat initial			Etat final			Entrées JK					
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	J_2	K_2	J_1	K_1	J_0	K_0
0	0	0	0	0	1	0	Ø	0	Ø	1	Ø
0	0	1	0	1	0	0	Ø	1	Ø	Ø	1
0	1	0	0	1	1	0	Ø	Ø	0	1	Ø
0	1	1	1	0	0	1	Ø	Ø	1	Ø	1
1	0	0	0	0	0	Ø	1	0	Ø	0	Ø

Tableau de Karnaugh et équations logiques

Bascule de poids 2²

J_2

Q_0 \ $Q_2 Q_1$	00	01	11	10
0	0	0	Ø	Ø
1	0	1	Ø	Ø

$J_2 = Q_1.Q_0$

K_2

Q_0 \ $Q_2 Q_1$	00	01	11	10
0	Ø	Ø	Ø	1
1	Ø	Ø	Ø	Ø

$K_2 = 1$

Bascule de poids 2²

J_1

Q_0 \ $Q_2 Q_1$	00	01	11	10
0	0	Ø	Ø	0
1	1	Ø	Ø	Ø

$J_1 = Q_0$

K_1

Q_0 \ $Q_2 Q_1$	00	01	11	10
0	Ø	0	Ø	Ø
1	Ø	1	Ø	Ø

$K_1 = Q_0$

Bascule de poids 2^0

J_0

Q_0 \ $Q_2 Q_1$	00	01	11	10
0	1	1	ø	0
1	ø	ø	ø	ø

$J_0 = \bar{Q}_2$

K_0

Q_0 \ $Q_2 Q_1$	00	01	11	10
0	ø	ø	ø	ø
1	1	1	ø	ø

$K_0 = 1$

Schéma logique (fig. 9.9)

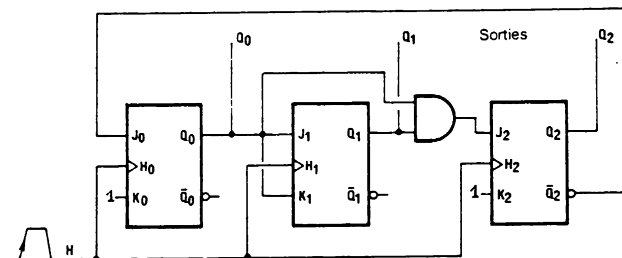


Fig. 9-9. Compteur synchrone module 5 (bascules JK).

Remarque :

L'étude du décompteur nécessite la même recherche et ne peut se déduire directement de celle du compteur.

2e exemple : compteur modulo 5 avec des bascules D.

Etat initial			Etat final			Entrées D		
Q_2	Q_1	Q_0	Q_2	Q_1	Q_0	D_2	D_1	D_0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	0	0	0	0	0	0

Tableaux de Karnaugh et équations logiques

Bascule de poids 2^2

D_2 — (Q_0 en lignes, Q_2Q_1 en colonnes)

Q_0 \ Q_2Q_1	00	01	11	10
0	0	0	ø	0
1	0	1	ø	ø

$$D_2 = Q_0 Q_1$$

Bascule de poids 2^1

D_1 — (Q_0 en lignes, Q_2Q_1 en colonnes)

Q_0 \ Q_2Q_1	00	01	11	10
0	0	1	ø	0
1	1	0	ø	ø

$$D_1 = Q_2\bar{Q}_0 + \bar{Q}_1 Q_0 = Q_0 \oplus Q_1$$

Bascule de poids 2^0

D_0 — (Q_0 en lignes, Q_2Q_1 en colonnes)

Q_0 \ Q_2Q_1	00	01	11	10
0	1	1	ø	0
1	0	0	ø	ø

$$D_0 = \bar{Q}_0 \bar{Q}_2$$

Schéma logique (fig. 9.10)

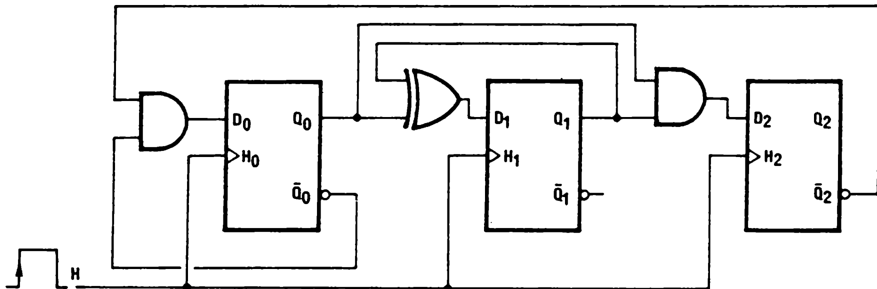


Fig. 9-10. Compteur synchrone modulo 5 (bascules D).

Remarques :

On peut remplacer la rétroaction synchrone précédemment décrite par une rétroaction asynchrone (si le compteur doit revenir à 0 après la valeur N, on décode N + 1 pour le remettre à 0, en agissant sur les entrées CLEAR asynchrones.

D'autre part, l'étude de compteurs binaires comptants dans une base autre que le binaire pur (Aiken, Excès de trois, Gray, etc.) est exactement la même que précédemment. On pourrait alors remplacer la rétroaction synchrone par une rétroaction asynchrone.

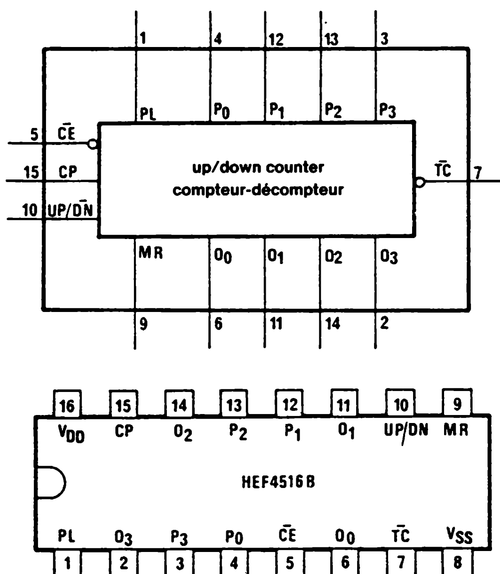
• Compteurs intégrés

Il existe sur le marché de nombreux compteurs intégrés binaires ou décimaux. Ils sont asynchrones ou synchrones. Ils sont en général à 4 bits mais peuvent avoir une capacité supérieure.

Ils peuvent avoir des entrées parallèles pour positionner le compteur à une certaine valeur, des sorties d'indication de valeur minimum ou maximum. Certains peuvent en outre compter ou décompter.

Exemples de compteurs intégrés CMOS

– 4516 : compteur/décompteur binaire 4 bits.



MR	PL	UP/$\overline{DN}$	$\overline{CE}$	CP	mode
L	H	X	X	X	parallel load ($P_n \rightarrow O_n$) chargement parallèle
L	L	X	H	X	no change - sans changement
L	L	L	L	∫	count-down, binary - décomptage
L	L	H	L	∫	count-up, binary - comptage
H	X	X	X	X	master reset - remise à zéro

H = HIGH state (the more positive voltage) (état haut : la tension la plus positive)
L = LOW state (the less positive voltage) (état bas : la tension la moins positive)
X = state immaterial (état indifférent)
⌠ = positive-going transition (front montant)

Logic equation for terminal count :
Equation logique de la retenue :

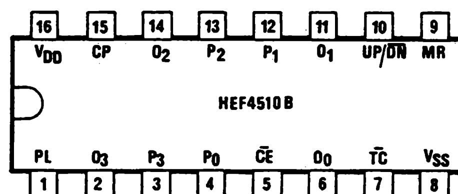
$$\left[\ \overline{TC} = \overline{\overline{CE} \cdot \{(UP/\overline{DN}) \cdot O_0 \cdot O_1 \cdot O_2 \cdot O_3 + (UP/\overline{DN}) \cdot \bar{O}_0 \cdot \bar{O}_1 \cdot \bar{O}_2 \cdot \bar{O}_3\}}\ \right]$$

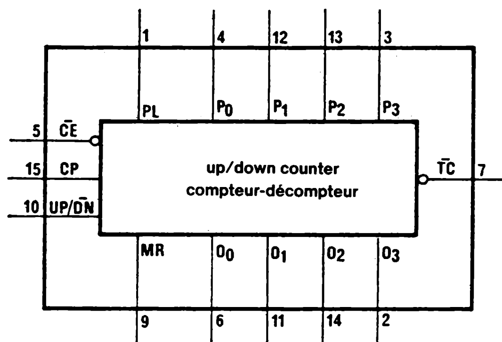
– 4510 : Décade de comptage-décomptage 4 bits (BCD)

MODE SÉLECTION

MR	PL	UP/$\overline{DN}$	$\overline{CE}$	CP	mode
L	H	X	X	X	parallel load ($P_n \rightarrow O_n$) chargement parallèle
L	L	X	H	X	no change - sans changement
L	L	L	L	∫	count-down, binary - décomptage
L	L	H	L	∫	count-up, binary - comptage
H	X	X	X	X	master reset - remise à zéro

H = HIGH state (the more positive voltage) (état haut : la tension la plus positive)
L = LOW state (the less positive voltage) (état bas : la tension la moins positive)
X = state immaterial (état indifférent)
⌠ = positive-going transition (front montant)





• Exemple d'interconnexion des décades 4510

Pour réaliser des compteurs de grande capacité, plusieurs compteurs sont associés suivant 2 modes :

Association mode asynchrone

La sortie de poids le plus fort (ou la retenue) de chaque compteur est le signal d'horloge du compteur suivant (fig. 9.11).
Ce type d'association est simple mais présente l'inconvénient des systèmes asynchrones (limitation de la fréquence de fonctionnement due au temps de propagation du signal d'horloge d'un compteur à l'autre).

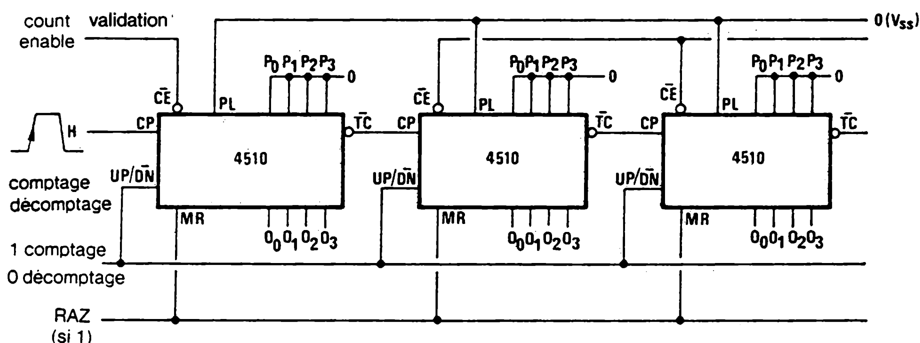


Fig. 9-11. Association de décades de comptage (mode asynchrone).

Remarque :

Même si elles ne sont pas utilisées, ne pas laisser en l'air les entrées P_0, P_1, P_2, P_3 (par exemple on les réunit au VSS (0)).

Association Mode synchrone

Tous les compteurs reçoivent le signal d'horloge.
Les entrées « d'autorisation » ($\overline{CE}$)et les sorties « retenue »($\overline{TC}$) permettent la liaison des compteurs dans la chaîne (fig. 9.12).
L'entrée « Autorisation » ($\overline{CE}$) du premier boîtier est reliée à la masse pour permettre le fonctionnement de celui-ci (poids les plus faibles), et la sortie « retenue » ($\overline{TC}$) du boîtier de rang i est reliée à l'entrée « Autorisation » ($\overline{CE}$) du boîtier de rang i + 1.

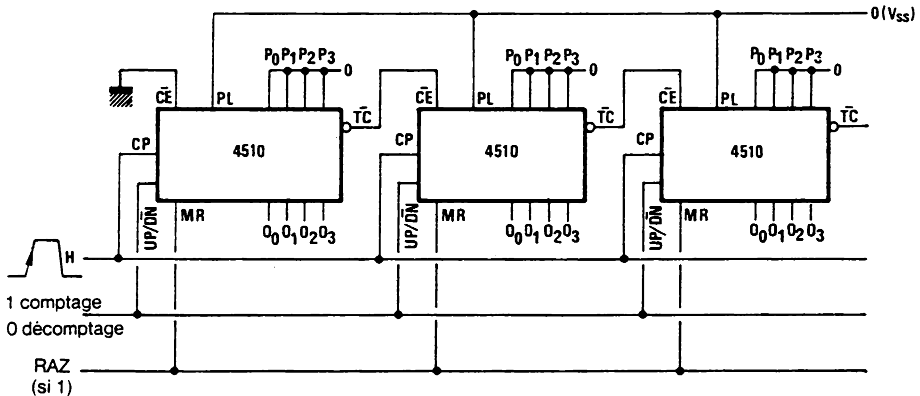


Fig. 9-12. Association de décades de comptage (mode synchrone)

• Exemples d'utilisation des entrées parallèles

Comptage de 0 à 8 avec une décade 4510

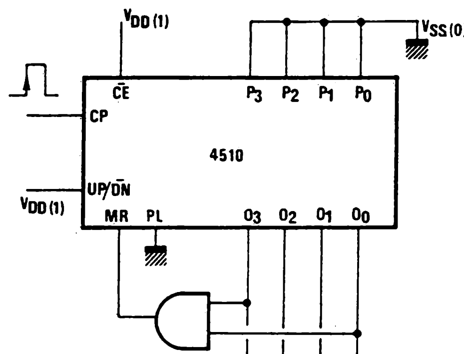


Fig. 9-13. Comptage à cycle incomplet avec une décade 4510.

Il suffit de « repérer » le chiffre 9 sur les sorties du compteur qui permettra la remise à 0 du compteur (activation de MR : Master Reset).
Si les sorties se nomment O_3 O_2 O_1 O_0, on aura 9 quand $O_3 = 1$ et $O_0 = 1$ donc $MR = O_3.O_0$ (MR = remise à zéro), d'où le schéma (fig. 9.13).

Comptage de 3 à 8 avec une décade 4510

Il faut décoder le chiffre 9 (8 + 1) sur les sorties du compteur qui permettra le chargement parallèle de la valeur 3 présente sur les entrées P_0 à P_3 (activation de PL : Parallel Load), d'où le schéma (fig. 9.14).

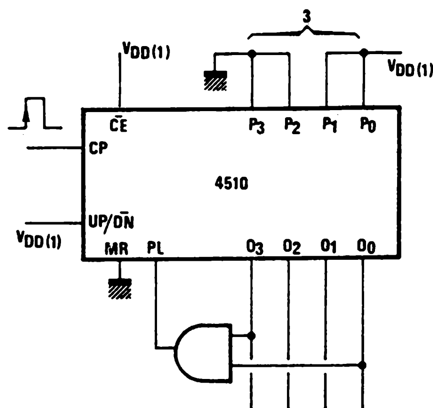


Fig. 9-14. Comptage de N_1 à N_2 avec une décade 4510.

Diviseur programmable avec une décade 4510

On désire obtenir un signal dont la fréquence est égale à la fréquence d'horloge divisée par N.

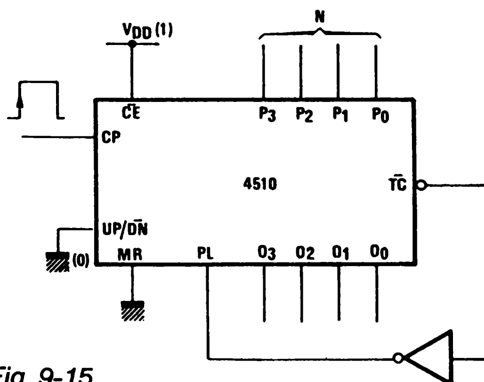
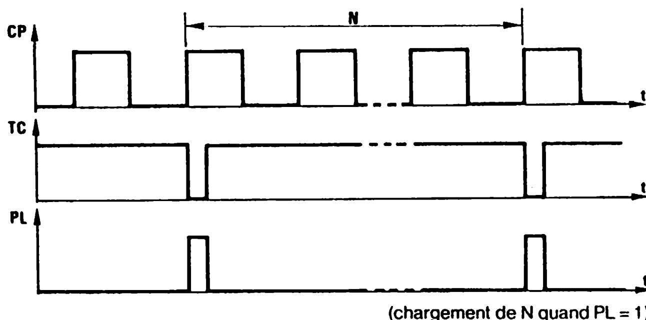


Fig. 9-15.

Il suffit pour cela de connecter la décade en décompteur (0 logique sur l'entrée UP/$\overline{DN}$) et de programmer la décade pour qu'elle puisse décompter de N à 0. (Le passage à 0 activera la sortie retenue $\overline{TC}$ qui agira sur l'ordre de chargement parallèle PL.) (fig. 9.15).
On aura donc la sortie $\overline{TC}$ qui passera à 0 toutes les N impulsions d'horloge.

Diagramme des temps



Générateurs de salves d'impulsions avec décade 4510 (fig. 9.16)

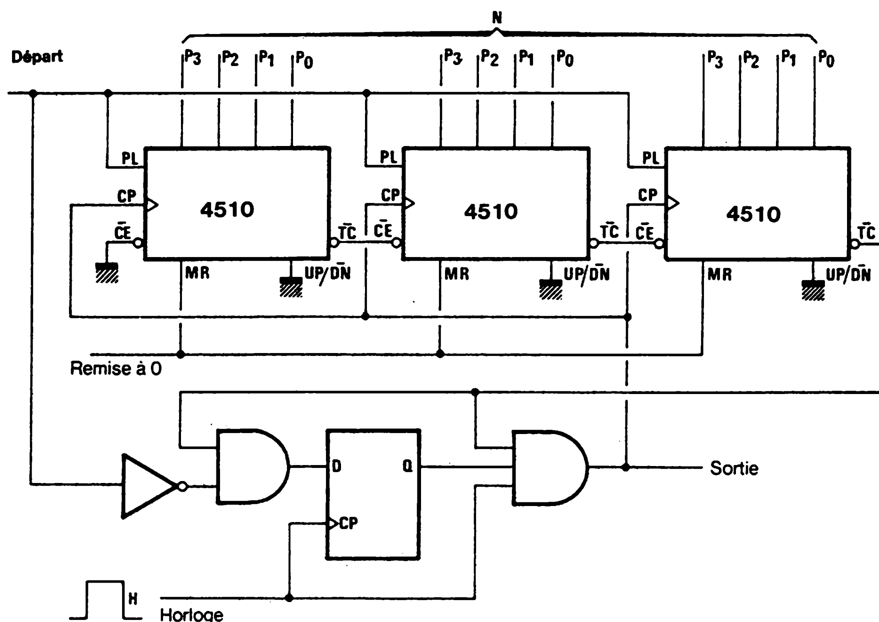
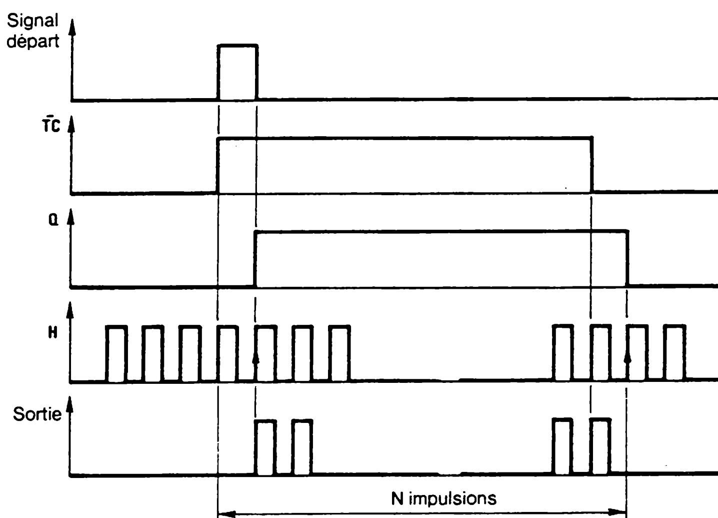


Fig. 9-16. Générateur de salves d'impulsions.

Lorsque le signal de départ est donné (impulsion), celui-ci agit sur le chargement parallèle des décades (PL = 1). A la retombée du signal de départ, les décades étant commandées en décompteur (UP/$\overline{DN}$ = 0), elles vont décompter de la valeur maximale affichée (chargement parallèle) à 0. Le signal de sortie est l'image de l'horloge tant que $\overline{TC}$ = 1, c'est-à-dire tant qu'on n'a pas atteint la valeur 0. On obtiendra donc en sortie une suite d'impulsions dont la valeur a été affichée avant le signal de départ.

Diagramme des temps



X - Mémoires

1. Introduction

Une mémoire est un dispositif permettant de **stocker des informations** (données). Lorsqu'on veut accéder rapidement à une donnée, celle-ci se trouvera à une **adresse** précise dans le dispositif, on utilisera ainsi une mémoire de type **accès aléatoire.** Lorsque cette rapidité n'est pas nécessaire, on pourra utiliser des mémoires de types **accès séquentiels.**
Les mémoires à accès aléatoires à semi-conducteurs pourront être de type **morte** c'est-à-dire à lecture seule ou de type **vive** c'est-à-dire à lecture et à écriture.
Les mémoires à accès séquentiels sont à lecture et écriture et peuvent être soit à semi-conducteurs (bulles magnétiques, registres CCD), soit magnétiques (disquette, disque, cassette, etc.) ; les bandes perforées et autres dispositifs mécaniques ont pratiquement disparus.
Le problème du stockage d'informations en grande quantité est résolu par l'emploi des mémoires de type séquentiel.

2. Type de mémoires à accès aléatoires

• Mémoires mortes : ROM, PROM, EPROM, etc.

Elles sont à lecture seule. Les informations ne sont pas perdues à la coupure de l'alimentation des circuits.

- **ROM :** (read-only-memory : mémoire à lecture seulement). Elle a été programmée par le fabricant de circuits intégrés par **masquage** et son programme ne peut être modifié.
- **PROM :** (programmable ROM : ROM programmable). Elle est livrée non-enregistrée par le fabricant. L'utilisateur programme sa PROM à l'aide d'un appareil spécial : le programmateur de PROM. Mais lorsque celle-ci est programmée, on ne peut plus l'effacer.
- **EPROM :** (erasable PROM : PROM effaçable)
  Par rapport à la PROM, l'avantage est que l'utilisateur, pourra s'il le désire, effacer son EPROM en la soumettant à un rayonnement ultra-violet intense.
- **EEROM :** (electrically erasable ROM : ROM effaçable électriquement). Ce sont les mémoires mortes les plus récentes. Elles sont programmables et effaçables électriquement par l'utilisateur.

• Mémoires vives : RAM (Random Access Memory : mémoire à accès aléatoire)

Elles sont fabriquées à partir des transistors bipolaires ou MOS, mais quelle que soit la technologie employée on trouve des RAM statiques ou dynamiques. Les informations seront de toute façon perdues à la coupure de l'alimentation des circuits.

- **RAM - Statiques :** La cellule de base (mémoire élémentaire) est la bascule. Il suffit donc de maintenir la tension d'alimentation du circuit sans autre précaution pour conserver l'information.
- **RAM - Dynamiques :** La cellule de base, dans les mémoires de type dynamique, est la capacité de grille-source d'un transistor MOS. Il est donc évident que si on ne veut pas perdre l'information (décharge de la capacité), il faut « rafraîchir » périodiquement cette cellule.
  Ces mémoires de structure plus simple que les statiques sont d'un emploi plus complexe car elles exigent des circuits de « rafraîchissement ».

3. Mémoires à accès séquentiel

Dans ces mémoires, on n'accède pas directement au mot cherché. Les informations doivent défiler jusqu'à ce qu'on arrive au point recherché.
C'est le cas de toutes les mémoires de masse comme la bande magnétique (cartouche ou cassette), les disquettes (Floppy Disk), les disques durs (dans les mini-ordinateurs) où on utilise le principe de l'aimantation rémanente d'une zone d'une bande magnétique.

Mais c'est le cas aussi des registres à décalage vus précédemment. A ce sujet, de nouvelles technologies sont actuellement développées, ce sont :

- **CCD** (Charge Coupled Device : circuit à couplage de charge) ou BBD (Bucket Brigade Device).
  Ce sont des registres à décalage qui permettent une plus grande densité d'intégration que les technologies MOS ou bipolaires. Leur fabrication est simple mais le fonctionnement est dynamique.
- **MBM** (Mémoires à bulles magnétiques)
  Ce sont les dernières nées, elles utilisent le phénomène de déplacement de bulles magnétiques dans certains matériaux. Leurs performances sont encore supérieures aux CCD ou BBD. Leur fonctionnement est statique.

Caractéristiques principales des mémoires

Capacité

C'est le nombre total de bits que le circuit peut emmagasiner. Les données se trouvent formatées en général en 1, 4 ou 8 bits, la capacité du circuit sera égale au nombre de mots total pouvant être enregistré, multiplié par le format (1, 4 ou 8).

Exemple

Une mémoire pouvant conserver 1 024 mots de 4 bits aura une capacité de $1\ 024 \times 4 = 4\ 096$ bits.

Remarque :

Si la mémoire conserve 2^n mots elle devra permettre l'accès à 2^n places différentes (adresses) celles-ci seront donc codées sur n bits (2^n combinaisons différentes). Ces n bits formeront le bus d'adresses.

Réponses de la mémoire

- *Temps d'accès :*
  (access time). Temps qui s'écoule entre l'instant où l'adresse et les signaux nécessaires au cycle de lecture sont appliqués à la mémoire et l'instant où la donnée se trouve en sortie.
- *Temps de cycle :*
  (cycle time). Temps minimum pour effectuer un cycle de lecture et un cycle d'écriture (RAM).

• Structure type d'une mémoire : (RAM) (fig. 10.1)

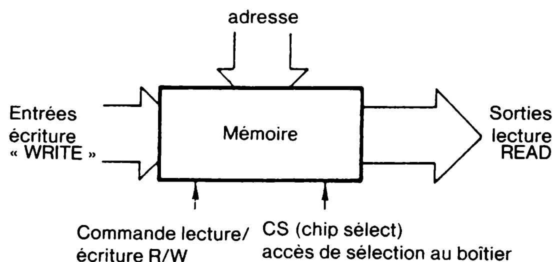
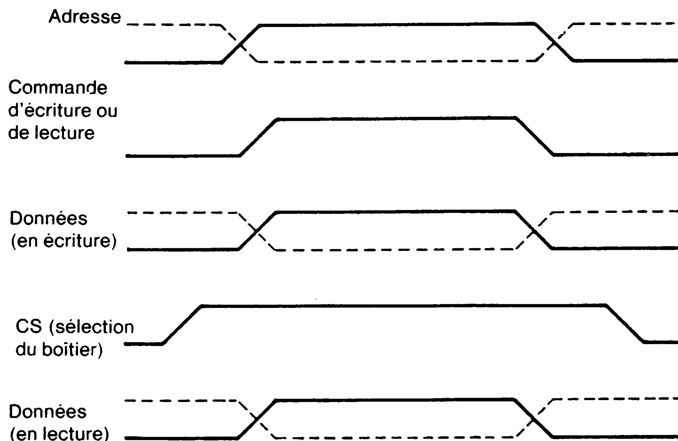


Fig. 10-1. Boîtier mémoire.

Remarque

Souvent, le bus de lecture et le bus d'écriture ne font qu'un : le **bus de données.**

• Principe fonctionnement (diagrammes types)



Ecriture d'une donnée : Pour inscrire une donnée dans la mémoire (si type RAM) il faut :

– **Appliquer l'adresse de l'emplacement** choisi sur le bus d'adresses.

– **Appliquer la donnée** de n bits sur les bus de données.

- **Activer la commande de sélection** du circuit (CS : chip select = 1 ou ($\overline{CS}$ = 0).
- **Activer la commande d'écriture** (W : Write = 1 ou R/$\overline{W}$ = 0).

Nota :

Le boîtier mémorisera le mot appliqué sur le bus de données à l'adresse désignée, que lorsque tout ce qui a été énoncé précédemment a été appliqué.

Lecture d'une donnée : Pour lire un mot dans la mémoire (type « vive » ou « morte », il faut :

- **Appliquer l'adresse de l'emplacement** où se trouve le mot (bus d'adresses)
- **Activer la commande de sélection** du circuit (CS ou $\overline{CS}$).
- **Activer la commande de lecture** (R : Read = 1 ou R/$\overline{W}$ = 1)
- **Relever l'information** (mot de n bits) sur le bus de données.

• Encapsulage des mémoires

Celles-ci se présentent le plus souvent en boîtier DIL dont le nombre de broches peut atteindre 64, en ce qui concerne les mémoires mortes ou vives.
L'utilisateur devra se reporter aux notices « constructeur » pour retrouver les **entrées-sorties** et **commandes** vues précédemment et leurs brochages.

• Capacité des mémoires

On a vu que l'on désigne par capacité le nombre maximum de bits que la mémoire peut emmagasiner. On utilise actuellement des mémoires de 512 kilobits (ROM) en technologie MOS. Mais on dépasse les 10^6 bits en technologie CCD et les 10^8 bits en MBM.

• Formatage des informations

Les mots enregistrés ont en général un format de 1 bit, de 4 bits ou de 8 bits (1 octet) pour la plupart des applications en micro-informatique, pour lesquelles le format des mots employés est de 8 ou 16 bits.

• Sortie 3 états

Plusieurs boîtiers pouvant être branchés sur les mêmes bus, leurs sorties peuvent prendre les états bas ou haut, soit 0 ou 1. Il est donc indispensable de pouvoir **déconnecter** les sorties qui ne sont pas en **travail.**

Cela est possible en permettant aux sorties d'être en état de haute impédance en plus de leurs états logiques normaux (0 à 1). Ceci est réalisé en connectant les sorties à des transistors qui se trouvent saturés lorsqu'un signal de commande apparaît.

Exemple avec les MOS (fig. 10.2).

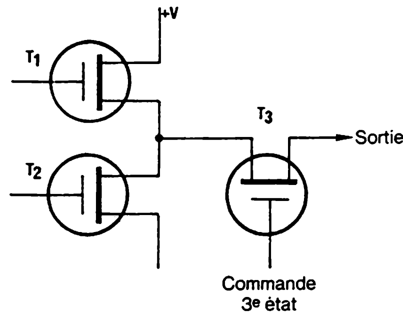


Fig. 10-2. Sortie 3 états.

5. Les ROM

• Principe

C'est le principe de la matrice à diodes (fig. 10.3)

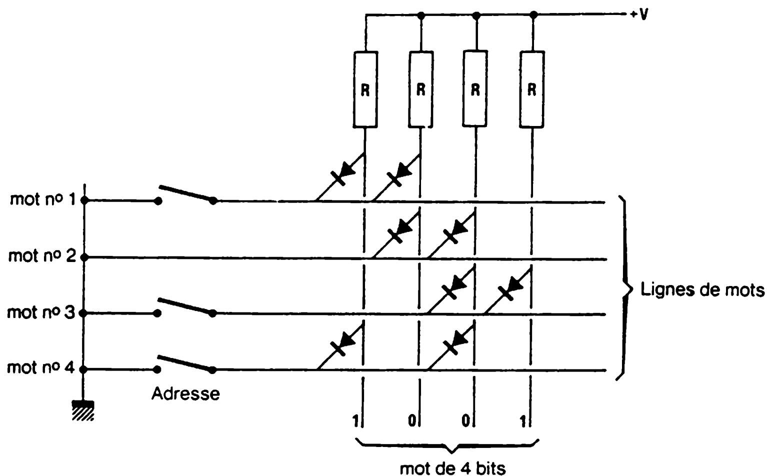


Fig. 10-3. Principe de la structure d'une ROM.

• Fonctionnement

C'est le mot n° 2 qui a été sélectionné (adresse), on peut lire sur les bus de sortie l'information 1001.

Remarque :

En réalité, l'interrupteur d'adressage est constitué par un interrupteur statique (transistor).

• Réalisation

On a vu précédemment dans les technologies à « diodes », que celles-ci présentaient quelques défauts. Pour les pallier, les constructeurs ont remplacé les diodes dans les matrices, par des transistors bipolaires au début, et par des transistors MOS ensuite. L'avantage de ce remplacement est que le transistor sert simultanément de connexion ligne-colonne et de commande pour l'adressage.

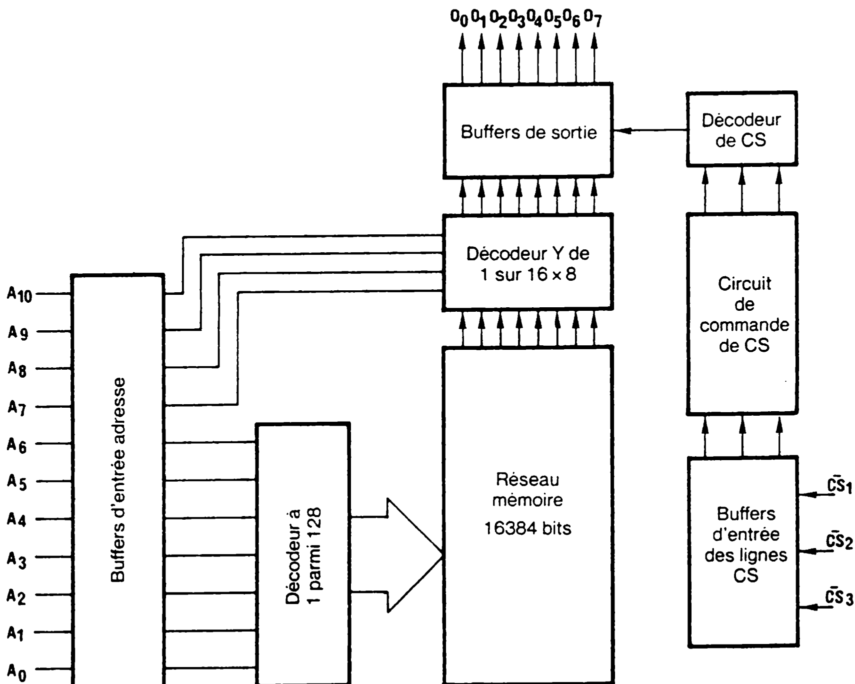


Fig. 10-4. Synoptique de la 2316 E de Intel.

• Exemple de circuits

– **2316 E de Intel :** Technologie MOS, 16 kilobits (2 048 × 8 bits) (fig. 10.4).

Remarque :

3 commandes de sélection de circuits ont été prévues. Elles doivent être appliquées simultanément pour que le circuit soit actif. De ce fait, ces trois lignes peuvent directement commander 8 ROM identiques.

Exercice :

Utiliser les 3 entrées de sélection $\overline{CS_1}$, $\overline{CS_2}$, $\overline{CS_3}$ pour commander 3 ROM identiques de type 2316 afin d'obtenir une capacité mémoire de 48 K bits.

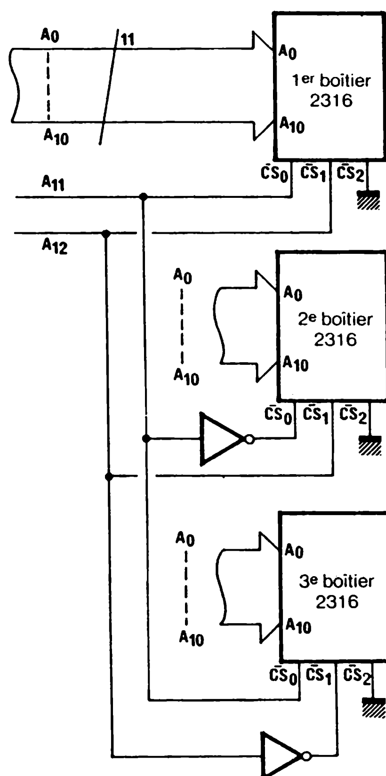


Fig. 10-5. Utilisation des commandes de sélection des 2316.

Solution :

Fonctionnement : Pour les 2 048 premières adresses ($A_{11} = A_{12} = 0$) le premier boîtier sera sélectionné ($\overline{CS_0} = A_{11}$; $\overline{CS_1} = A_{12}$).
Pour les 2 048 suivantes ($A_{11} = 1$, $A_{12} = 0$), c'est le deuxième qui le sera ($\overline{CS_0} = \overline{A_{11}}$; $\overline{CS_1} = \overline{A_{12}}$)
Pour les 2 048 suivantes ($A_{11} = 0$, $A_{12} = 1$), c'est le troisième qui sera sélectionné puisque $\overline{CS_0} = \overline{A_{11}} = 0$ et $\overline{CS_1} = \overline{A_{12}} = 0$ (fig. 10.5).

Conclusion : La sélection a pu se faire simplement en utilisant des portes logiques de base.

• Applications

L'utilisation a été vue précédemment, elle est simple : on applique une **adresse** et on **lit** la donnée se trouvant à cette adresse. On va voir quelques applications :

Mémoires de programmes : système à microprocesseur.

En micro-informatique, on sépare les mémoires des données des mémoires de programmes. La mémoire de programmes est figée et logée dans une ROM. Par contre, la mémoire de données – qui sont des variables qui interviennent dans le programme – est logée dans une mémoire à lecture-écriture, une RAM (fig. 10.6).

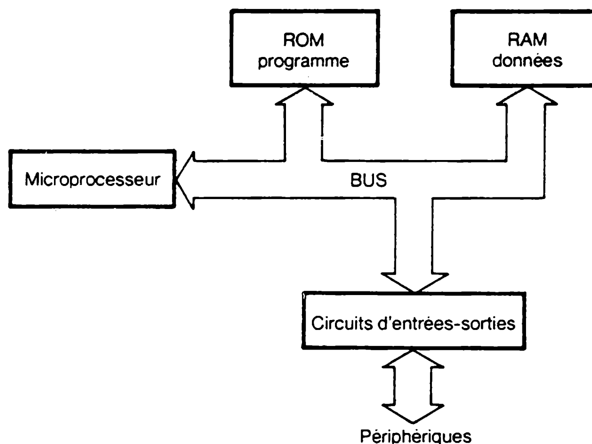


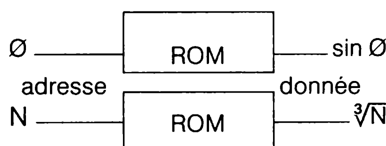
Fig. 10-6. Principe microsystème ou organisation d'un système à microprocesseur.

Microprogrammation

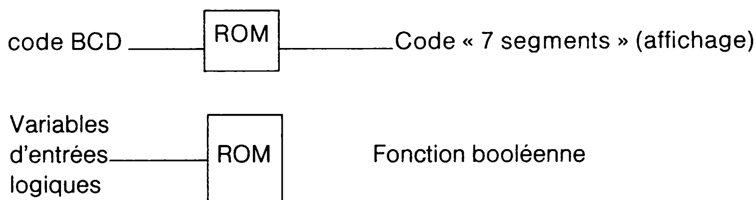
Selon le principe de la microprogrammation **l'instruction** en langage machine est appliquée au microproccesseur.
Cette instruction va déclencher le microprogramme (séquence de micro-instructions) qui en assure l'exécution.
Ce microprogramme est enregistré dans une ROM à demeure dans la puce « microprocesseur ».

Utilisation en transcodeurs

On fait correspondre à l'adresse une information et à la donnée une fonction de cette information ; ainsi peut-on réaliser des transcodeurs mathématiques comme :



On peut réaliser aussi des transcodeurs pour des applications logiques comme :



Le principal avantage dans ces cas est le gain considérable en temps pour obtenir le « résultat » en comparaison avec les méthodes séquentielles de calcul.

Conclusions

- La ROM est une mémoire non-volatile.
- On ne peut plus par la suite modifier le « programme » enregistré.
- Si on désire une programmation du circuit, il faut la commander au fabricant qui en facturera l'étude du masquage et la réalisation.
- Intéressant si fabrication de circuits en grande série (calculatrices, jouets, micro-ordinateurs grand public, etc.).

6. LES PROM

• Principe

L'avantage des PROM sur les ROM est que ce n'est plus le fabricant qui « enregistre » la mémoire mais l'utilisateur. L'inconvénient est que la PROM une fois enregistrée ne peut pas être « effacée » (comme la ROM).

Programmation

La PROM est livrée vierge, c'est-à-dire qu'elle se trouve avec des « 1 » ou des « 0 » partout.
L'utilisateur devra, selon son programme, inscrire l'inverse là où il le désire.

Il existe deux technologies donc deux méthodes de programmation :
- **Destruction de fusibles** pour les PROM à fusibles (fusibles au nichrome ou au silicium).
- **Court circuitage de jonctions** pour les PROM à jonctions.

• PROM à fusibles

Ce sont les plus répandues. On ajoute en série avec chaque diode (supposons un réseau à diodes comme les ROM) un fusible que l'on pourra laisser intact ou faire fondre suivant la programmation (fig. 10.7) ·
- si on laisse le fusible intact, la diode est connectée entre colonne et rangée,
- si l'on détruit le fusible (à l'aide d'une forte impulsion de courant), colonne et rangée sont isolées.

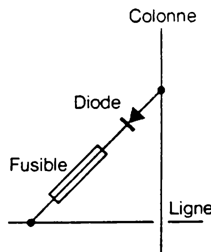


Fig. 10-7. Point mémoire d'une PROM à fusibles.

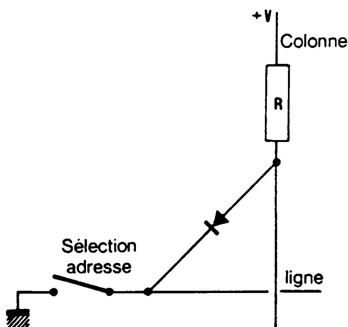


Fig. 10-8. Si l'adresse est sélecttionnée, on lit sur la colonne : 0. Le fusible n'a pas été détruit.

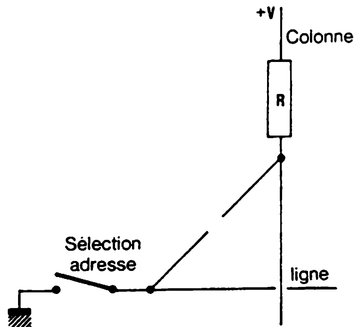


Fig. 10-9. Si l'adresse est sélectionnée, on lit sur la colonne : 1 Le fusible a été détruit.

Lors de la programmation, le courant nécessaire pour la fusion du fusible est de quelques dizaines de milliampères.
Les « fusibles » des premières PROM étaient réalisés à base d'un alliage de nickel et de chrome (fusibles au nichrome). Cet alliage présentait un défaut : après destruction, le fusible se reformait parfois par corrosion (phénomène de « grow-back ») ce qui modifiait, bien entendu, les informations enregistrées dans la PROM.
On est vite arrivé aux PROM à fusibles au silicium polycristallin qui ne présentent pas le défaut précédent. Un courant d'environ 25 mA est nécessaire pour la destruction de ce fusible.

Remarque

La programmation se fait avec un appareil que l'on appelle « le programmateur de PROM ». Son rôle est d'appliquer à la mémoire, aux adresses désignées, des **impulsions** de courant par le bus de données qui **détruiront** les fusibles (connexion rétrécie en nichrome-aluminium ou silicium polycristallin) là où nous voudrons modifier les états des points de connexion de la matrice.
Selon les types de mémoire, la valeur de l'impulsion de courant varie de 10 mA à 1 A.

• PROM à jonctions

Le principe est l'inverse de la PROM à fusibles.
C'est-à-dire que la modification de l'état logique d'un point de connexion se fera en établissant un contact là où il n'y en avait pas.

Principe (fig. 10.10)

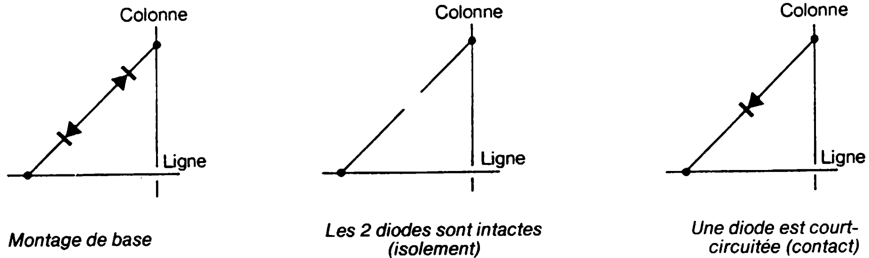
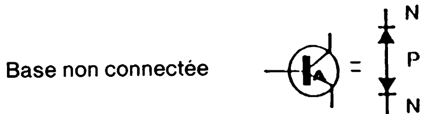


Fig. 10-10. Point mémoire de la PROM à jonctions.

Remarque

Pratiquement ces diodes sont constituées par les jonctions d'un transistor bipolaire NPN dont la base est « en l'air ».



Programmation

Cette technologie a été développée par INTERSIL, elle est appelée : **Migration par avalanche induite.**

Le principe en est le suivant (fig. 10.11) : si une forte impulsion de courant est appliquée entre **Emetteur** et **Collecteur** du transistor dont la coupe est donnée, il y a dans la jonction **Base Emetteur** un phénomène d'avalanche qui se produit. Il y a donc élévation de température sur un étroit canal qui fait fondre l'aluminium court-circuitant la jonction.

La flèche représente la migration de l'aluminium

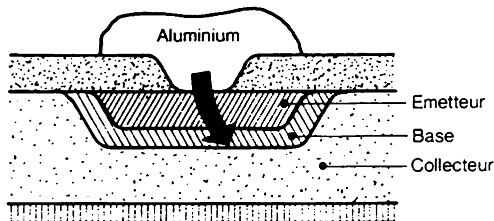


Fig. 10-11. Coupe du transistor d'une PROM à jonctions.

Exemples de circuits

- PROM 0512 de Harris à fusibles : 64 mots de 8 bits.
- PROM 6610/6612 de Harris : technologie CMOS : 256 mots de 4 bits.

Applications

Ce sont les mêmes que les ROM.

Conclusions

- Ce sont des mémoires **non-volatiles.**
- Les PROM permettent une rapide implantation logicielle, lors de mise au point de systèmes puisque la programmation est faite par l'utilisateur.
- La solution « PROM » revient moins cher que la solution « ROM » (pour de petites quantités).
- Elles se comportent comme des ROM, les modifications sont impossibles.
- Il faut prévoir un programmateur de PROM (demande un investissement de quelques milliers de francs).

7. Les EPROM (ou REPROM)

• Principe

Une EPROM est une mémoire morte qui se programme à l'aide d'un programmateur de PROM. Mais soumise à un rayonnement ultraviolet, elle s'efface.

Le point mémoire est réalisé à l'aide de transistor FAMOS.

Transistor FAMOS :

Principe : le transistor FAMOS (Floating gate Avalanche Injection Mos)

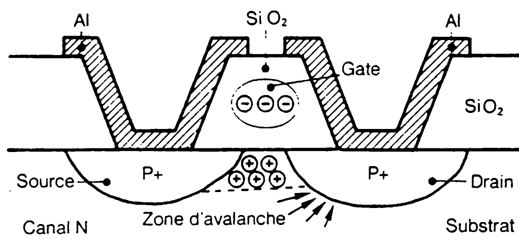


Fig. 10-12. Coupe du transistor FAMOS.

est un transistor MOS canal P dans lequel la grille (gate) n'est reliée à aucun conducteur (fig. 10.12).

Fabrication :

Sur un substrat de type N faiblement dopé, une couche d'oxyde de silice est déposée. On réalise ensuite deux ouvertures pour permettre la diffusion d'impuretés d'atomes accepteurs ce qui dope très fortement (P+) ces deux zones qui après métallisation (Aluminium) deviennent le drain et la source.

La grille flottante est constituée par une armature en silice qui a été « piégée » dans l'oxyde de silicium lorsque celui-ci a été déposé.

Fonctionnement :

Si on soumet les électrodes Source-Drain à une tension impulsionnelle, cela provoque un pic de courant, fortement négatif, par rapport au drain.

Un effet d'avalanche se produit entre le drain (P) et le substrat (N) et par effet TUNNEL, des électrons sont injectés dans la grille (flottante). Lorsque l'impulsion disparaît, les électrons qui se trouvent piégés dans la grille, (qui n'est reliée nulle part), ne peuvent s'écouler tout seuls.

C'est comme si le transistor était commandé, c'est-à-dire que la grille « polarisée » produit un canal P et le transistor est passant (sortie logique = 0).

Nota :

C'est pour ces raisons que la lecture d'une EPROM « vierge » ne donne que des 1 (transistor FAMOS bloqué) et le programmateur de PROM « écrira » des 0 dans les cellules que l'on voudra modifier.

Lorsqu'on soumet l'EPROM à un rayonnement ultraviolet intense, la charge stockée s'écoule en quelques dizaines de minutes (dépend de la puissance du rayonnement), car les électrons piégés reçoivent une énergie suffisante pour franchir l'isolant (S_iO_2).

La longueur d'onde du rayonnement ultraviolet est de 2 537 Å et une intensité lumineuse de 1 200 $\mu W/cm^2$, ce qui permet un effacement complet en 15 mn environ.

On estime que le transistor une fois saturé (c'est-à-dire l'EPROM programmée) et non-soumis à un rayonnement ultraviolet garderait 80 % de sa charge sur la grille au bout de 100 ans à 125 °C de température ambiante.
On estime la densité du courant de décharge à 10^{-40} A/cm^2.

Présentation :

On reconnaît une mémoire EPROM car son boîtier (type DIL en général) présente sur le dessus une fenêtre en quartz pour que le rayonnement UV parvienne aux transistors FAMOS.

• Exemples de circuits

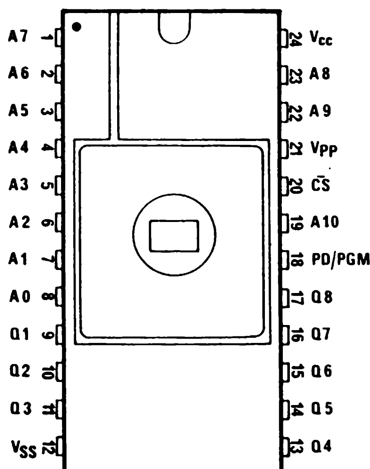
- EPROM 1701 de Intel : 2 048 bits (256 × 8 bits)
- EPROM 2516 : 16 kilobits (2 048 × 8 bits)
- EPROM 2532 : 32 kilobits (4 096 × 8 bits)
- EPROM 2564 : 64 kilobits (8 192 × 8 bits)

• Etude de deux EPROM classiques : la 2516 et la 2532

Caractéristiques générales :

- 2516 : 2 K × 8 bits.
- 2532 : 4 K × 8 bits.
- Alimentation unique 5 V.
- Existence d'une ROM compatible broche à broche avec la 2532.
- Fonctionnement statique.
- Faible consommation :
  2516 : 285 mW en fonctionnement et 50 mW en veille,
  2532 : 400 mW et 50 mW.
- Grande immunité aux bruits

BROCHAGE DE LA 2516 ET DE LA 2532



Broches	Fonctions
A_0 - A_{09} (A_{11} pour la 2532)	Adresses
$\overline{CS}$	Sélection boîtier
PD/PGM ($PD/\overline{PGM}$ pour la 2532)	Commande basse consommation et programmation
Q_1 - Q_8	Entrées/Sorties
Vcc	+ 5 V
Vpp	+ 25 V (en programmation)
Vss	0V (masse)

Pour la 2532 :
Broche 18 : A11
Broche 20 : $PD/\overline{PGM}$

Description :

Les 2516 et 2532 sont des EPROM de 16 384 et 32 768 bits.
Elles sont fabriquées en MOS canal N. Les entrées et sorties de chaque EPROM sont compatibles TTL.
Les sorties sont à 3 états pour l'utilisation d'un bus commun.
En programmation, des signaux TTL de durée 50 ms sont nécessaires ainsi qu'une tension de 25 V.
Il faut compter 100 s pour programmer entièrement la 2516 et le double pour la 2532.

Conditions de fonctionnement:

Fonction et numéro de la broche du circuit (Vcc à + 5 V et Vss à 0 V)						
Mode Fonctionnement volts	Type du circuit	2516	PD/PGM (18)	Vpp (21)	$\overline{CS}$ (20)	Qi (9 à 11 et 13 à 7)
		2532	PD/$\overline{PGM}$ (20)	Vpp (21)		Qi (9 à 11 et 13 à 17)
Lecture	2516		0	+ 5	0	Sorties
	2532		0	+ 5		Sorties
Inhibition circuit lors de la lecture	2516		Indiff.	+ 5	+ 5	Haute impédance
	2532		+ 5	+ 5		Haute impédance
Inhibition circuit faible consommation	2516		+ 5	+ 5	Indiff.	Haute impédance
	2532		+ 5	+ 5		Haute impédance
Programmation	2516		+ 5 50 ms 0	+ 24	+ 5	Entrées
	2532		+ 5 50 ms 0	+ 24		Entrées
Inhibition circuit lors de la programmation	2516		0	+ 24	+ 5	Haute impédance
	2532		+ 5	+ 24		Haute impédance
Vérification programmation	2516		0	+ 24	0	Sorties

Lecture :

Les informations sont disponibles sur les sorties Q (Q_1 – Q_8) avec un retard de 450 ns (150 ns pour la 2516).
Si CS = 1 (2516) ou PD/$\overline{PGM}$ = 1 (2532), les sorties de la mémoire sont dans le 3e état (haute impédance).

Consommation :

Celle-ci peut être réduite (hors fonctionnement) de 80 % si PD/PGM ou PD/$\overline{PGM}$ est au niveau logique 1. (voir tableau de fonctionnement).

Programmation :

Au départ, tous les bits sont à 1. Quand Vpp = 25 V et que $\overline{CS}$ = 1 (2516), les données présentes en parallèle sur les broches Q_1 à Q_8 sont mémorisées si PD/PGM reçoit une impulsion 0 – 5 V – 0 pendant 50 ms (pour la 2532, le signal est inversé : impulsion de 5 V – 0 – 5 V).

Effacement :

Les mémoires sont effacées à l'aide d'une lampe émettant de fortes radiations ultraviolettes (la longueur d'onde est de = 2 537 Å).
L'effacement complet de la mémoire s'effectuera en 20 mn environ.

Blocage programmation : (inhibition lors de la programmation).

Si plusieurs EPROM sont connectées en parallèle, il est possible de les programmer simultanément. Mais on peut si on le désire bloquer la programmation pour certaines d'entre elles en appliquant 0 sur PD/PGM (2516) ou 5 V (2532) et 5 V sur $\overline{CS}$ de la 2516.

Vérification programmation :

On peut, après une programmation de la 2516, vérifier le contenu de la mémoire sur les sorties Q en appliquant OV sur $\overline{CS}$ mais en laissant le 24 V sur Vpp (Vérification pendant la programmation).

Conclusions

Avantages :

- Mémoires **non-volatiles.**
- Mémoires **effaçables** (mais en bloc seulement).
- Il existe généralement une ROM ayant la **même capacité** et le **même brochage.**

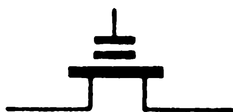
Inconvénients :

- L'effacement sélectif est impossible.
- Le prix de l'EPROM est supérieur à celui d'une ROM.
- Il faut s'équiper d'un matériel de programmation et d'effacement d'EPROM.

8. Les EEROM (ou EAROM)

• Principe

On a cherché à éviter l'effacement des mémoires « reprogrammables » par les UV pour des raisons de facilité d'utilisation et de rapidité. Les progrès de la technologie ont permis la mise au point d'un transistor spécial, le MNOS qui se sature exactement de la même façon que le transistor FAMOS (mémoires EPROM), mais qui se bloque par effet TUNNEL lors d'un effacement sous l'effet d'une impulsion électrique.



Symbole du MNOS

• Exemple de circuit : EEROM 5213

La 5213 est une EEROM de 2 K × 8 bits qui fonctionne avec une tension de 5 V lors de la lecture et de l'écriture d'un octet. Une application pendant 5 ms de la donnée est nécessaire lors de la programmation du circuit.
Cette opération ainsi que l'effacement d'un octet peut être répétée 10 000 fois. Le temps d'accès pendant la lecture est inférieur à 250 ns. Le temps d'effacement d'un octet ou du circuit complet est inférieur à 10 ms.

Brochage et désignation des broches (fig. 10.13).

$A_0 - A_{10}$: Bus d'adresses.

$\overline{CE}$: Sélection du circuit ; $\overline{OE}$: Sélection des sorties.

$\overline{WE}$: Sélection programmation (idem Vpp des EPROM).

I/O : Entrées des données lors de l'écriture ou de l'effacement. Sorties des données lors de la lecture.

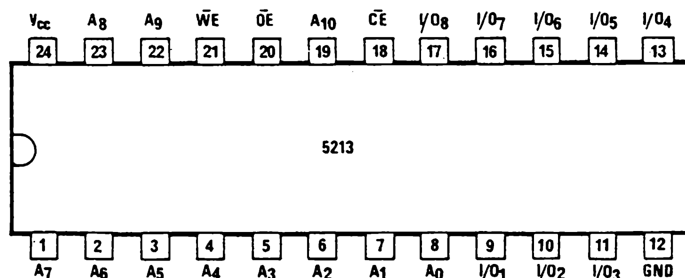


Fig. 10-13. Brochage de la 5213.

Tableau de fonctionnement de la 5213

Fonction et numéro broche				
Mode de Fonctionnement	$\overline{CE}$ (18)	$\overline{OE}$ (20)	$\overline{WE}$ (21)	I/O (9-11) (13-17)
Lecture	0 V	0 V	+ 5 V	Sorties
Inhibition (lors de la lecture	+ 5 V	Indiff.	+ 5 V	Haute impédance
Effacement d'un octet (10 ms)	0 V	+ 5 V	0 V	Entrées (+ 5 V)
Programmation de l'octet (5 ms)	0 V	+ 5 V	0 V	Entrées
Effacement du circuit (10 ms)	0 V	+ 15 V	+ 15 V	Entrées (+ 5 V)
Inhibition lors de la programmation ou de l'effacement	+ 5 V	Indiff.	Indiff.	Haute impédance

• Applications

On utilise l'EAROM dans tous les systèmes où l'on est contraint de modifier, relativement souvent, des paramètres dans un programme comme :

- Générateur de caractères programmable.
- Caisses enregistreuses.
- Pompes à essence.
- Taxiphones.
- Systèmes de péage sur autoroutes, etc.

Les EAROM sont de plus en plus utilisées en remplacement des EPROM.

On les utilise aussi actuellement en remplacement des RAM qui sont volatiles (si le temps de l'écriture (10 ms) ne pose pas de problèmes pour l'exécution du programme).

9. Les RAM

• Définition

C'est une mémoire à lecture et à écriture où l'on peut accéder directement à tous ses points mémoires grâce à leurs adresses.

On distingue deux types de RAM :

- les RAM statiques,
- les RAM dynamiques.

• RAM statique

Le point mémoire est réalisé avec une bascule (Flip Flop) (fig. 10.14.a et fig. 10.14.b).

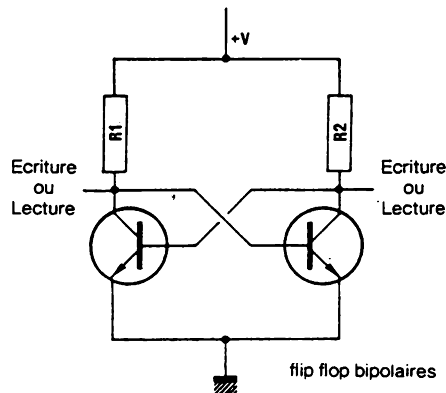


Fig. 10.14.a. Point mémoire d'une RAM bipolaire (principe)

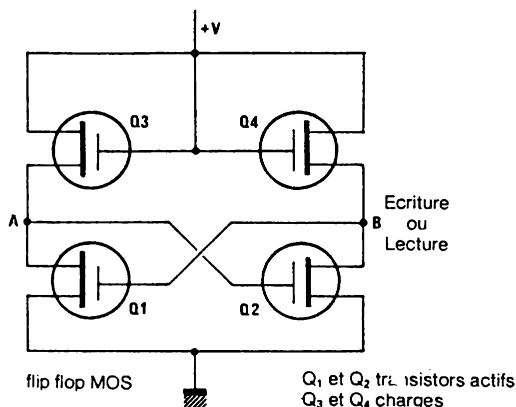


Fig. 10-14.b. Point mémoire d'une RAM Mos (principe)

Remarque :

Une cellule fondamentale comprend en réalité 6 transistors avec les deux transistors supplémentaires d'accès : Q_5 et Q_6 (fig. 10.15).

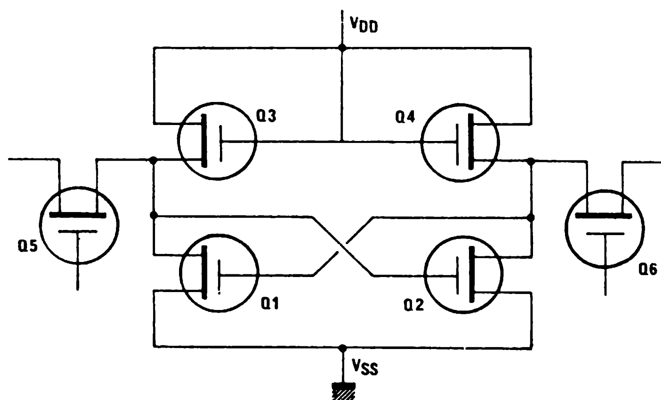


Fig. 10-15. Point mëmoire d'une RAM (réalisation).

Exemples de RAM statiques :

- 2114 (sortie 3 états) de Intel : 4 kilobits (1 024 × 4 bits).
- 6164 (sortie 3 états) : 64 kilobits (8 192 × 8 bits).

Exemple de brochage d'une RAM classique : la 2114 (fig. 10.16)

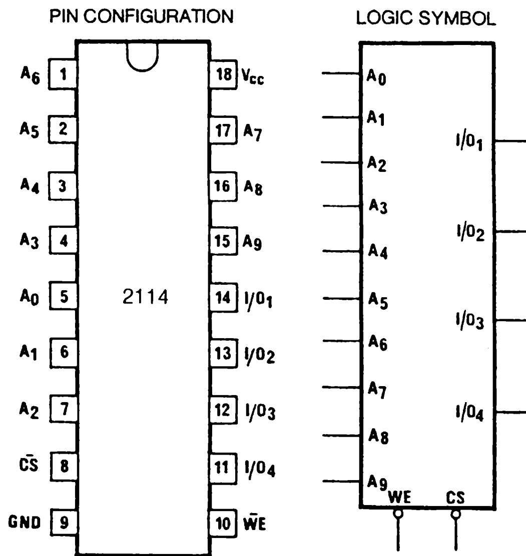


Fig. 10-16. Brochage de la 2114.

Désignation des broches : (2114)

A_0 à A_9 : Bus d'adresse

I/O_1 à I/O_4 : Entrée/Sortie des données.

$\overline{WE}$: Sélection de l'écriture

$\overline{CS}$: Sélection du circuit.

Tableau de fonctionnement : (2114)

Fonction et numéro broche			
Mode de fonctionnement	$\overline{CS}$ (8)	$\overline{WE}$ (10)	I/O (11 à 14)
Lecture	0 V	+ 5 V	Sorties (données)
Ecriture	0 V	0 V	Entrées (données)
Inhibition	+ 5 V	Indiff.	Haute impédance

• RAM dynamique

Le principe est que chaque point mémoire est réalisé autour d'un condensateur chargé (niveau 1) ou déchargé (niveau 0). Pour éviter de perdre l'information (décharge du condensateur), un système va lire le niveau de tension aux bornes du condensateur qui le rechargera éventuellement (si lecture d'un 1). C'est le processus de rafraîchissement que réalisera ce système.

Avantage :

Chaque point mémoire n'est réalisé qu'avec seulement un composant : le condensateur.

Nota :

En fait, ce condensateur est l'une des capacités de structure du transistor MOS.
Avec la RAM dynamique, la densité d'intégration est plus importante qu'avec la RAM statique (1 transistor au lieu de 6).

Inconvénient

La RAM dynamique ne peut être utilisée qu'avec son circuit de rafraîchissement.

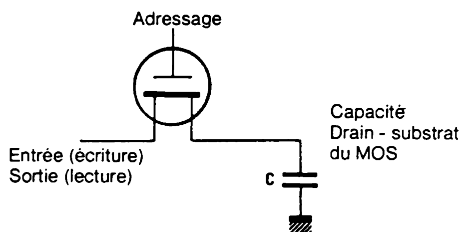


Fig. 10-17.

Le MOS est adressé par sa grille et l'on peut charger le condensateur (écriture) ou lire son niveau de tension (lecture) car le MOS est bidirectionnel (fig. 10.17).

Exemple de RAM dynamiques :
- 4146 : 16 kilobits (16 384 × 1 bit)
- 4454 : 256 kilobits (65 536 × 4 bits)

• Applications des RAM

Les principales applications des RAM se trouvent dans les systèmes programmés (micro-informatique, informatique, etc.).

Elles servent de mémoires pour les tables de données, les variables, les compteurs. Pour savoir s'il est préférable d'utiliser des RAM dynamiques plutôt que des RAM statiques, il faut savoir qu'une RAM dynamique possède une densité d'intégration plus importante qu'une RAM statique mais qu'elle demande un circuit de rafraîchissement. D'autre part, pour des applications en informatique industrielle où l'on va rencontrer des milieux relativement parasités, il est préférable d'éviter les mémoires dynamiques qui ont une moins bonne immunité aux bruits que les mémoires statiques.

Pour les autres applications, c'est la capacité mémoire qui nous guidera dans notre choix.

- **Jusqu'à 64 K octets,** il est préférable d'utiliser des RAM statiques organisées en mots de 8 bits ou 4 bits (montage en parallèle pour 1 octet).
- **Au-delà de 64 K octets,** il vaut mieux recourir à des RAM dynamiques car la surface occupée par les boîtiers sur le circuit imprimé est moindre qu'avec des RAM statiques et les circuits de rafraîchissement sont ici amortis.

Remarque :

Il existe des microprocesseurs comme le Z 80 qui sont dotés de circuits de rafraîchissement internes. Dans ce cas, il est plus intéressant d'utiliser des RAM dynamiques.

10. Les plans mémoires

Il est toujours possible d'associer des circuits intégrés mémoires (ROM, PROM, RAM, etc.) afin de réaliser l'extension de leur capacité, horizontalement (longueur d'un mot) ou verticalement (nombre de mots).

Exemple 1

Soit à organiser un plan mémoire de 4 096 octets à partir de 2 048 octets d'EPROM et 2 048 de RAM avec un circuit EPROM et un circuit RAM.
On désire en outre que les adresses basses affectent les EPROM et les adresses hautes les RAM.

Principe adressage : chaque boîtier, ayant une capacité de 2 K octets, recevra les 11 fils d'adresse ($2^{11} = 2$ K) de poids les plus faibles du bus d'adresse.
La capacité totale étant de 4 K octets, le douzième fil d'adresse ($2^{12} = 4$ K) permettra la sélection de l'un des deux boîtiers.

Quand on aura ce douzième fil, que l'on nommera AD_{11} par exemple si le premier fil se nomme AD_0 ; à la valeur logique 0 on sélectionnera le premier boîtier c'est-à-dire l'EPROM et quand A 11 sera à la valeur logique 1 il devra sélectionner le deuxième boîtier, c'est-à-dire la RAM. Le schéma logique qui en découle est évident.

Schéma logique : (fig. 10.18)

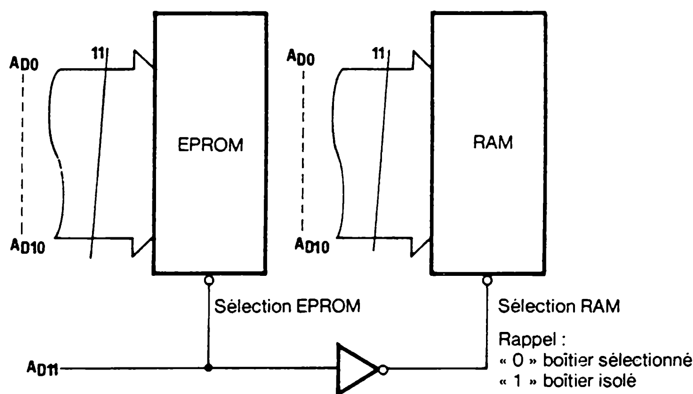


Fig. 10-18.

Exemple 2

Faire le schéma d'un plan mémoire dont les caractéristiques sont les suivantes :

- Capacité 20 K organisée en 4 K de RAM et 16 K d'EPROM.
- Mémoires utilisées : + EPROM type 2732 (4 K octets)
  + RAM type 2114 (1 K × 4 bits).
- Pagination « programmable » par straps dans un espace mémoire de 64 K (microsystème délivrant 16 fils d'adresses : $2^{16} = 64$ K).

Indications :

On pourra utiliser le comparateur arithmétique 7485 qui délivre une sortie « A = B » à 1 quand les 2 mots A et B présentés en entrées sont égaux.
D'autre part, on utilisera des décodeurs du type 74155 (double 2 vers 4 ou simple 3 vers 8).

Principe adressage :

Pour la partie EPROM, 4 boîtiers seront nécessaires (4 K × 4) pour obtenir les 16 K nécessaires. Chaque circuit recevra les 12 fils d'adres-

ses de poids les plus faibles (2^{12} = 4 K) et la sélection de l'un des circuits parmi 4 pourra se faire à l'aide d'un décodeur binaire à 2 entrées -4 sorties (1 parmi 4), ce décodeur étant alimenté par les 2 fils d'adresse du poids immédiatement supérieur, c'est-à-dire AD 12 et AD 13. Ainsi, on aura utilisé 14 fils d'adresses ce qui nous donne bien 16 K adresses différentes (2^{14} = 16 K).

Si on désire « programmer » la pagination de notre plan mémoire, autrement dit décider à quel endroit de l'espace mémoire de 64 K se trouvera les 16 K d'EPROM (4 places possibles : 16 × 4 = 64), il est par

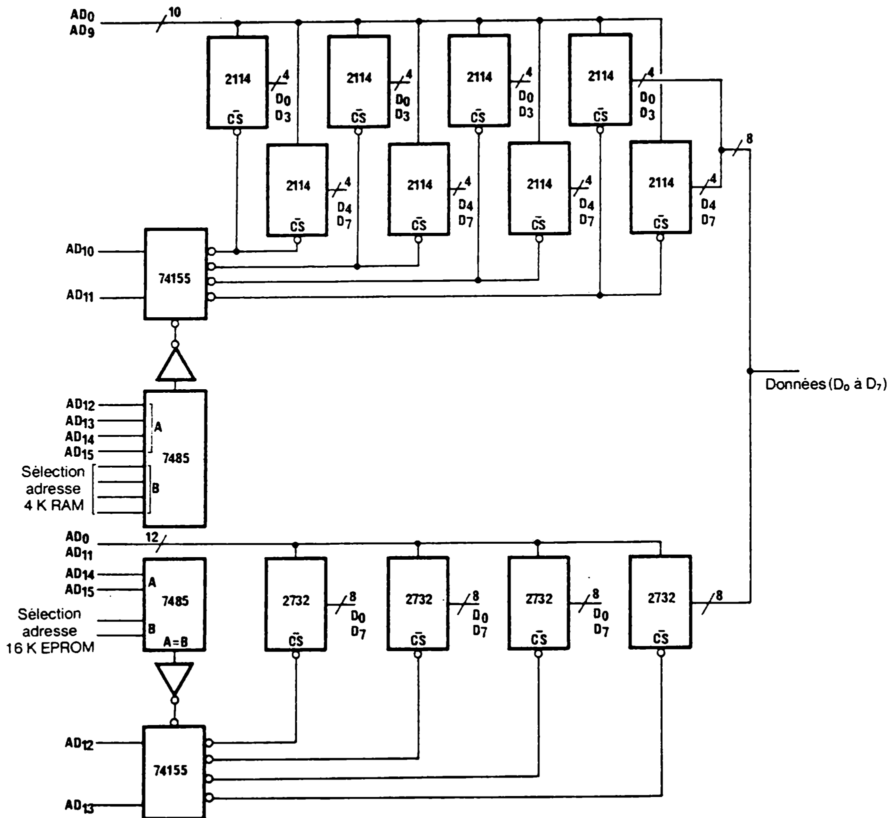


Fig. 10-19. Plan mémoire de 20 K organisé en 16 K d'EPROM et 4 K de RAM.

exemple possible d'utiliser des comparateurs arithmétiques comme le 7485 étudié précédemment qui délivre une sortie active quand il y a identité de 2 mots à l'entrée du circuit (le 7485 compare 2 mots de 4 bits).

On pourrait donc utiliser ce comparateur en alimentant la première entrée avec les 2 fils de poids les plus forts du bus d'adresse (AD 14 et AD 15) (l'entrée possédant 4 bits, les 2 non utilisées pourraient, par exemple, être mis à la masse) et l'autre entrée avec des niveaux logiques modifiables par straps (fig. 10.19).

Pour la partie RAM le principe est le même, le plan étant organisé à l'aide de 8 boîtiers de 1 K × 4 bits puisqu'on désire 4 K octets de RAM, les boîtiers seront regroupés 2 par 2 pour obtenir à chaque adresse sélectionnée, un octet (fig. 10.19).

Note : on peut utiliser des boîtiers de plus grande capacité comme l'EPROM 2764 (8 K octets) et la RAM 6116 (2 K octets). A capacité égale on emploiera moins de boîtiers, mais le principe du décodage demeure le même.

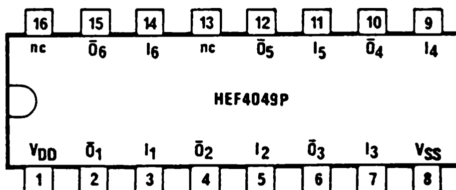
XI - Circuits d'adaptation et de traitement des signaux

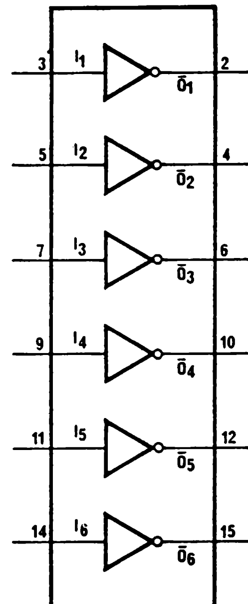
On rassemble ici un certain nombre d'exemples de circuits rencontrés fréquemment dans les systèmes logiques.

1. Adaptation CMOS - TTL

On peut utiliser pour l'adaptation CMOS-TTL des buffers (amplificateurs de charge). Ce sont des circuits qui permettent d'avoir une sortance plus importante dans une même famille logique.

Exemple de circuit :
CMOS 4049, 4050 (fig. 11.1)





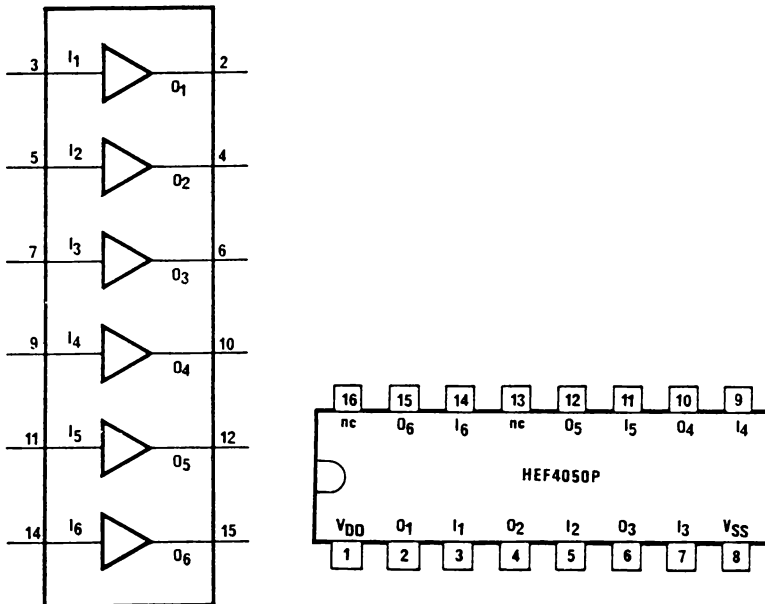


Fig. 11-1. Brochage des circuits 4049 et 5050.

Sortance garantie

Type de circuit alimenté	Nombre de circuits
TTL standard	2
TTL type 74 LS	9
TTL type 74 L	16

Autres exemples de circuits : TTL 7406

6 portes NON : Courant de sortie admissible par porte 40 mA.

Rappel : Le courant de sortie d'une porte pour la TTL standard est de 16 mA (au niveau logique 0).

2. Interfaces d'usage général

• Introduction

Il existe de nombreuses applications qui mettent en œuvre des circuits intégrés logiques associés à des transistors. Ce sont, par exemple, les circuits d'interface entre une logique TTL et des lampes, des relais, des thyristors, des triacs, etc.

• Spécifications

Ce sont des circuits qui comportent en général une porte CMOS ou TTL en entrée et un transistor à la sortie.
On leur demande :

- Une fréquence élevée de fonctionnement (10 MHZ par exemple).
- Une compatibilité avec des tensions couramment utilisées (5 à 48 V).
- Une compatibilité en entrée avec les niveaux logiques (5 V en général).
- Un fort courant de sortie (100 mA par exemple).

Exemple de circuits :

- SN 75450 et SN 75460 de « Texas Instrument » (fig. 11.2).

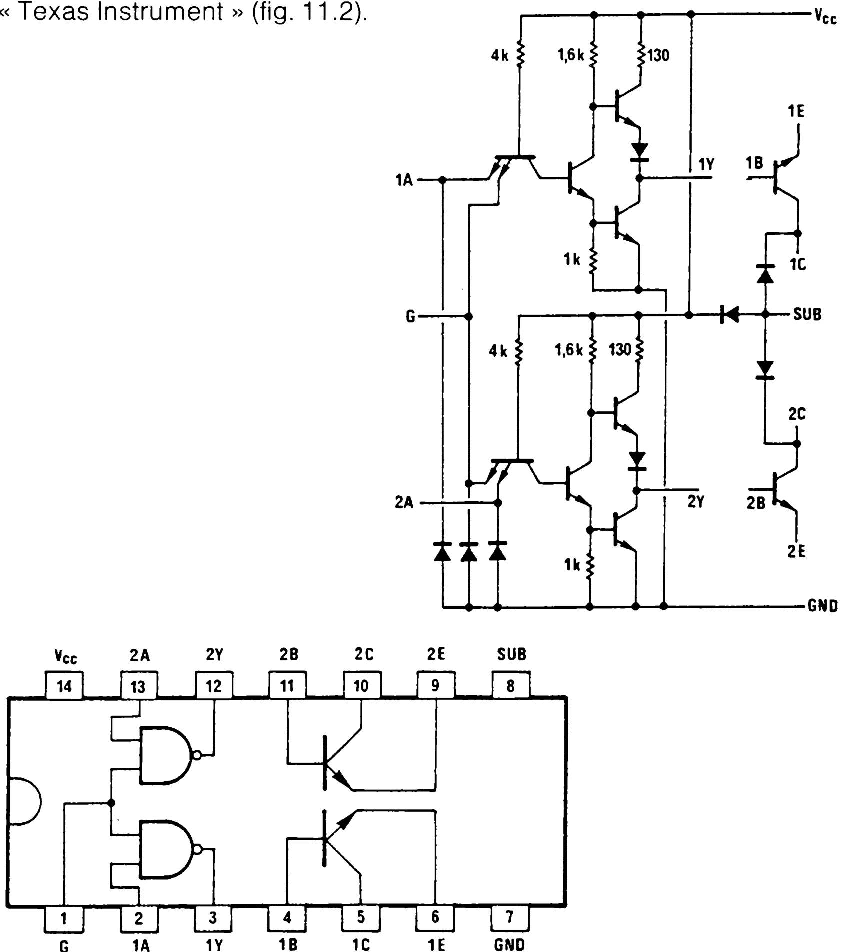


Fig. 11-2. Circuit SN 75450 de T.I.

3. Trigger de Schmitt

• Principe

C'est un circuit qui traite les signaux logiques à front lents en permettant une mise en forme de ces signaux.

• Propriétés et symboles

Représentation d'un inverseur « Trigger de Schmitt » (fig. 11.3).

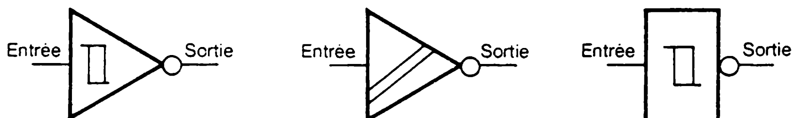


Fig. 11-3.

Caractéristiques Entrée-Sortie de l'inverseur Trigger de Schmitt (fig. 11.4).

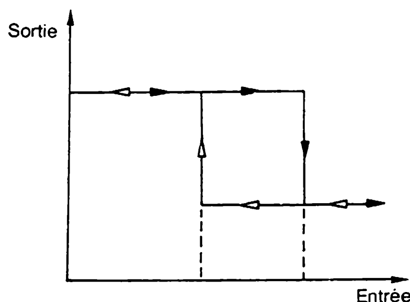


Fig. 11-4.

Conséquence sur la mise en forme d'un signal quelconque d'entrée (fig. 11.5).

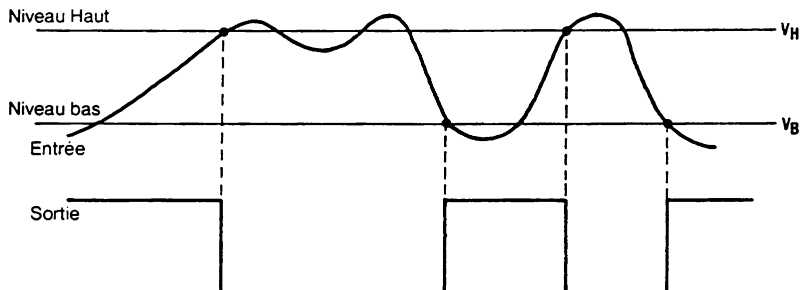


Fig. 11-5.

• Applications

– **Mise en forme d'un signal :**
Le signal e provient par exemple d'un système de mesure qui délivre une information qui n'est pas compatible (signal lent) avec les circuits logiques traditionnels.
Un Trigger de Schmitt permettra l'interfaçage nécessaire.

– **Réalisation d'une horloge** à 100 Hz à partir du secteur (fig. 11.6).
On peut après une mise en forme du signal redressé double alternance à partir du secteur 50 Hz, obtenir une base de temps à 100 Hz synchronisé avec ce secteur.

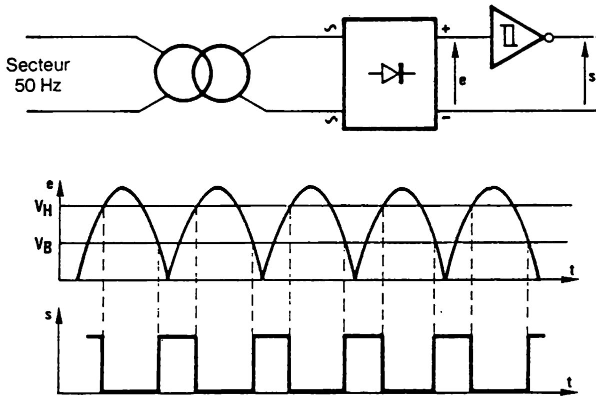
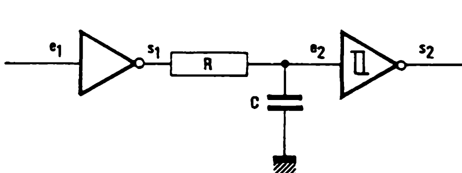
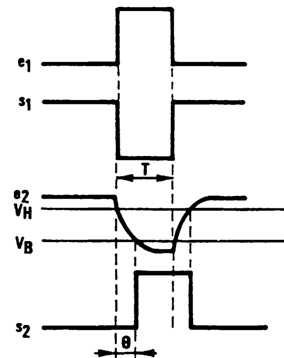


Fig. 11-6. Schéma et signaux obtenus.

– **Circuit retard :** (fig. 11.7)





– **Fonctionnement**
Condition à satisfaire

$T > 4\,RC$

Fig 11-7. Retard d'un signal.

On a obtenu le retard θ entre le signal d'entrée et le signal de sortie.

– **Oscillateur :** la fréquence du signal obtenu est fonction de la constante de temps RC et de l'hystérésis du Trigger de Schmitt (fig. 11.8).

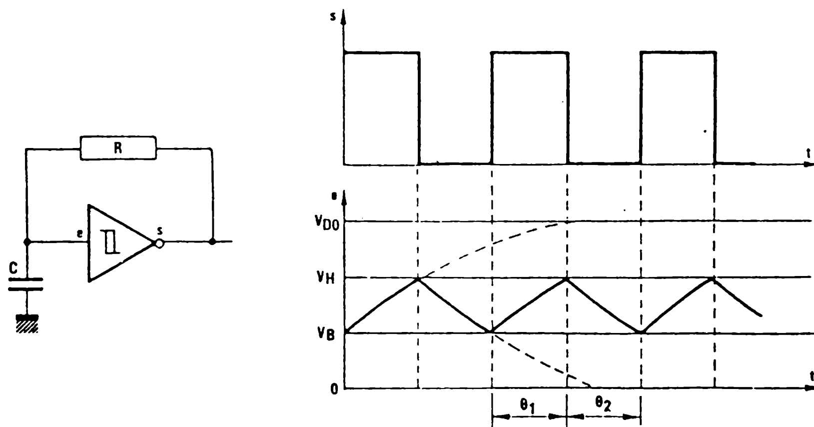


Fig. 11-8. Oscillateur à Trigger de Schmitt.

Exemple :

Circuit 4093 P CMOS (4 portes NAND Trigger)
si $V_{DD} = 15$ V
$V_B = 5{,}4$ V
$V_H = 8{,}9$ V

Calculer T en fonction de la constante de temps RC.
Calculons le temps θ_1 nécessaire pour atteindre V_H partant de V_B (fig. 11.8).

On trouve :

$$\theta_1 = -\,RC \text{ Log } \frac{V_{DD} - V_H}{V_{DD} - V_B}$$

Calculons le temps θ_2 nécessaire pour atteindre V_B partant de V_H.

On trouve :

$$\theta_2 = RC \text{ Log } \frac{V_H}{V_B}$$

$$\text{et } T = \theta_1 + \theta_2 = RC \text{ Log}\left[\frac{(V_{DD} - V_B)}{(V_{DD} - V_B)} \cdot \frac{V_H}{V_B}\right]$$

$$\text{A°N° : } T = RC \text{ Log}\left[\frac{(15 - 5{,}4)}{(15 - 8{,}9)} \cdot \frac{8{,}9}{5{,}4}\right]$$

$T \simeq = 0{,}953$ RC

Remise à 0 d'un système logique :

A la mise sous tension, la capacité C se comportant comme un court-circuit, la tension d'entrée du Trigger de Schmitt est nulle, la sortie est égale à 1.

La capacité C se charge à travers R sous la tension Vcc et lorsque le niveau de basculement V_h du Trigger est atteint, la sortie retombe à 0 (fig. 11.10).

Le système logique a donc reçu, à la mise sous tension, une impulsion de remise à zéro (initialisation, « reset », etc.).

Remarque :

Il est possible de prévoir une remise à zéro du système en fonctionnement, pour cela il suffit de placer un bouton-poussoir « RAZ » en paral-

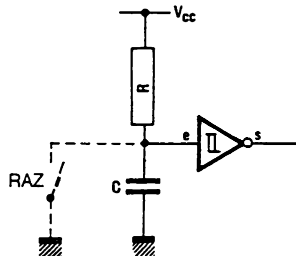


Fig. 11-9. Remise à 0 d'un système logique.

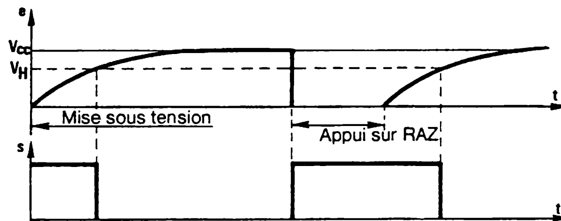


Fig. 11-10. Diagrammes de fonctionnement.

lèle avec le condensateur C pour obtenir à l'appui un niveau logique 1 à la sortie et au relâchement le maintien à 1 calibré par la constante de temps RC (fig. 11.10).

• Exemples de circuits intégrés

TTL : 74132 4 - TRIGGER (fig. 11-11)
CMOS : 4093 P 4 NAND - TRIGGER (fig. 12-12)

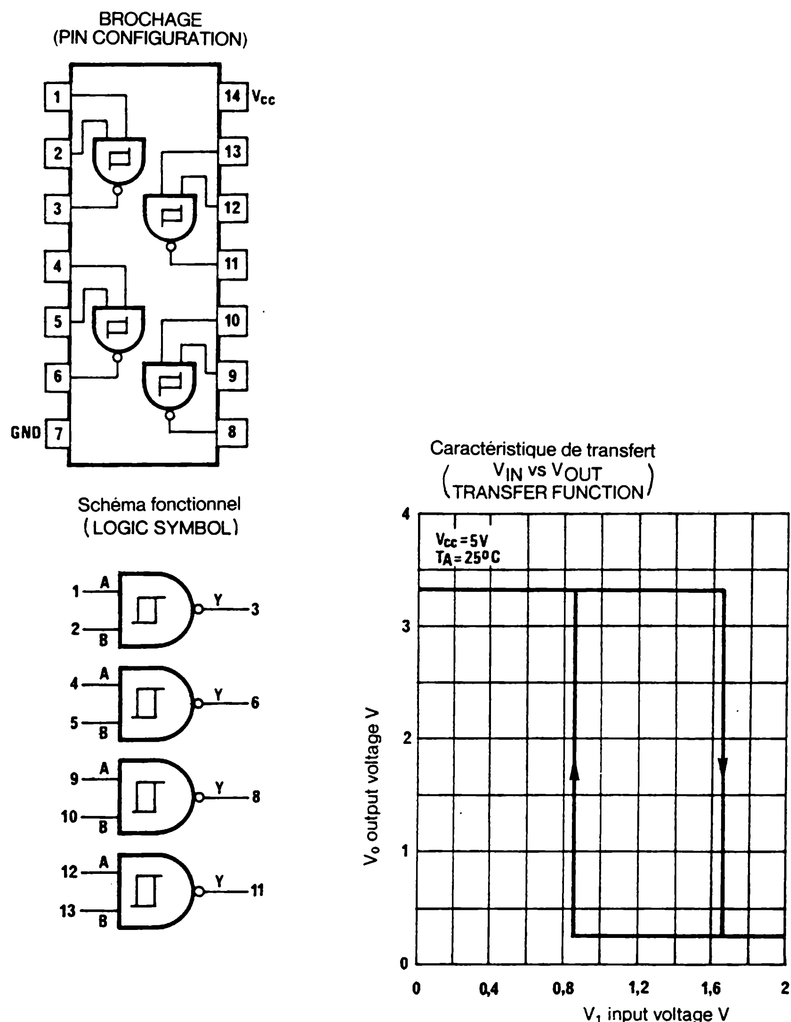
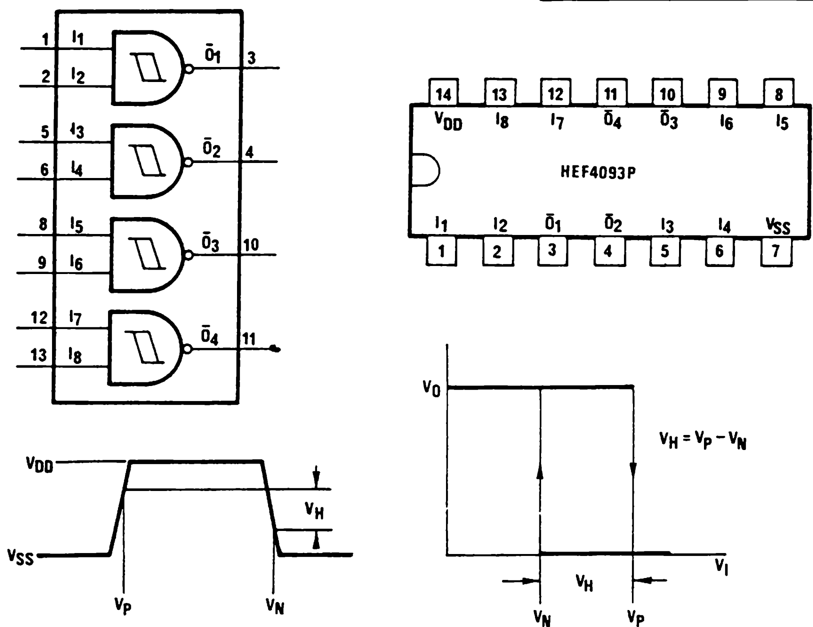


Fig. 11-11. Brochage et courbe de transfert du circuit 74132.



Signaux montrant la définition de V_P ; V_N ; V_H
Waveform showing definition of V_P ; V_N ; V_H ;

Caractéristique de transfert
Transfert characteristics of 1 or 5 gates

Niveaux de commutation :	Symbol	Tamb (°C) -40 typ.	25 min.	25 typ.	25 max.	85 typ.		V_{DD} (V)
Tension d'entrée Front montant (Positive trigger threshold voltage)	V_P	3,0	2,3	2,9	3,5	2,9	V	5
		5,9	4,5	5,9	7,0	5,9	V	10
		–	–	8,9	–	–	V	15
Tension d'entrée Front descendant (Negative trigger threshold voltage)	V_N	2,6	1,5	2,3	2,7	2,1	V	5
		4,0	3,0	3,9	5,5	3,8	V	10
		–	–	5,4	–	–	V	15
Tension d'hystéresis (Hysteresis voltage)	V_H	0,4	0,4	0,6	–	0,8	V	5
		1,9	1,0	2,0	–	2,1	V	10
		–	–	3,5	–	–	V	15

Fig. 11-12. Brochage et caractéristique du 4093.

4. Monostable

Le monostable délivre en sortie une impulsion calibrée par un circuit RC qui lui est associé. Cette impulsion est déclenchée par un changement de niveau à l'entrée du monostable.

• Diagrammes et symbole : (fig. 11.13)

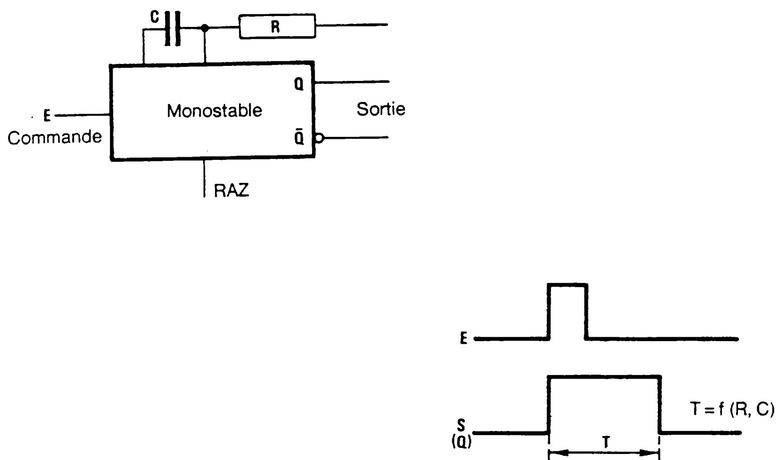


Fig. 11.13: Représentation et diagrammes de fonctionnement d'un monostable.

• Remarque

Il existe deux types de monostable :

- Monostable réamorçable.
- Monostable non-réamorçable.

Les différents diagrammes montrent les différences entre les deux types de monostables.

• Comparaison entre les deux types de monostables

Si la période t du signal d'entrée est supérieure à la largeur T donnée par le réseau RC (fig. 11.4). Le signal de sortie a les mêmes caratéristiques que le monostable soit réamorçable ou non.

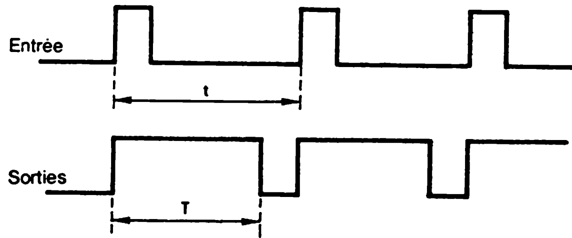


Fig. 11-14.

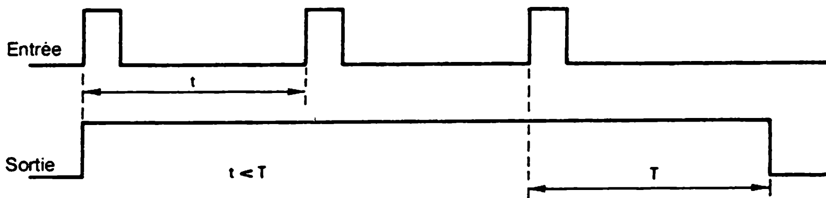


Fig. 11-15. Monostable réamorçable (diagrammes).

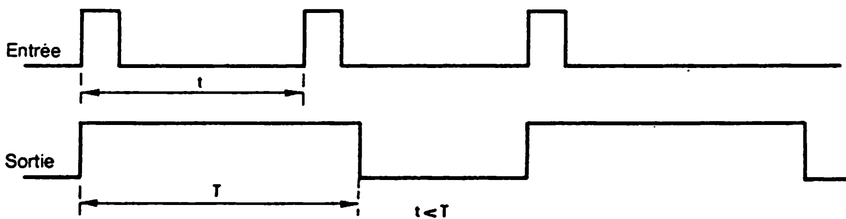


Fig. 11-16. Monostable non réamorçable (diagrammes).

Si la période t du signal d'entrée est inférieure à la largeur T donnée par le réseau RC, les 2 types de monostable donnent des résultats différents (fig. 11.15 et 11.16).

• Applications

Chaque fois que l'on veut obtenir des signaux carrés avec une durée déterminée à partir de signaux de durées quelconques, on utilisera des monostables.

Exemples

Obtention d'impulsions à partir d'un signal carré de rapport cyclique quelconque.
Obtention d'un signal carré à partir d'impulsions de courte durée.

Nota :

La durée T est donnée par le constructeur soit sous forme d'une équation fonction de R, C et de la tension d'alimentation, soit sous forme d'abaque.

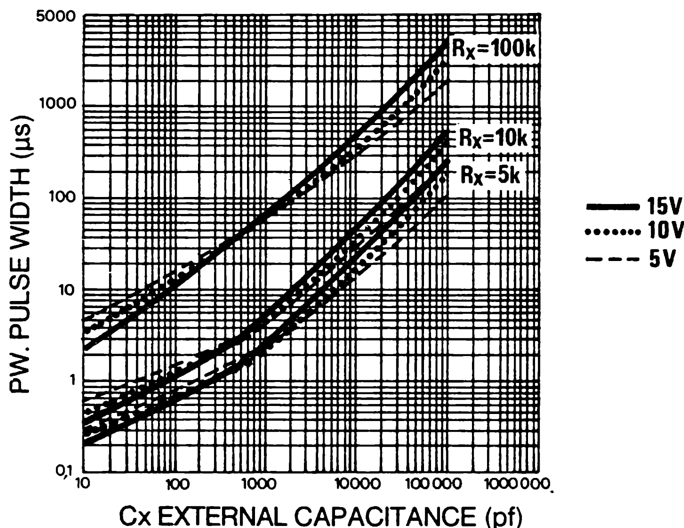
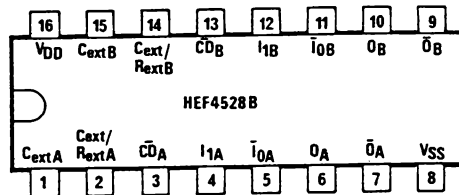


Fig. 11-17. Exemple d'abaque donnée par le constructeur.

• Exemples de circuits

– **TTL :** 74121 N non-réamorçable 1 monostable
74122 N réamorçable 1 monostable.

– **CMOS :** 4528 B réamorçable 2 monostables (fig. 11.18).



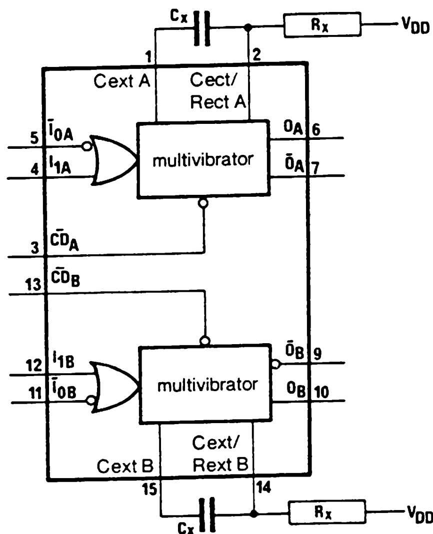


Fig. 11-18. Représentation et brochage du 4528.

XII - Conversion analogique-numérique et numérique-analogique

1. Information analogique et information logique

Une information analogique est une information dont la valeur varie de manière continue dans le temps. L'intérêt de l'information analogique est qu'elle peut représenter une grandeur physique avec une infinité de valeurs intermédiaires. Les inconvénients majeurs de l'information analogique sont d'une part l'impossibilité d'un traitement numérique de celle-ci et d'autre part la distorsion possible du signal causé par des bruits quelconques générés par les systèmes.

D'autre part, une information numérique constituée d'un certain nombre de 0 et 1 (mot binaire) a l'inconvénient de ne pouvoir représenter une grandeur physique que d'une manière discontinue mais a l'avantage de pouvoir être traitée par des calculateurs numériques et ce signal est peu sensible aux bruits et distorsions.

C'est pour ces raisons que l'on va préférer transformer une information analogique en information numérique pour un traitement de celle-ci à l'aide de microsystèmes et éventuellement retransposer le résultat numérique en information analogique dans la chaîne du système si un récepteur doit être commandé ainsi (moteur, table traçante, etc.)

Ce sont ces circuits qui transforment un signal analogique en signal numérique ou inversement, que l'on va étudier : on les appelle des **convertisseurs.**

2. Convertisseurs analogiques-numériques (C.A.N)

• Principe

La conversion consiste à rechercher une expression numérique dans un code donné pour représenter une information analogique avec une précision et une résolution données.
Nous allons étudier les techniques principales utilisées dans la conversion analogique-numérique.

• Conversion par simple rampe

Un générateur de tension délivre à une entrée d'un comparateur un signal à dents-de-scie. L'autre entrée du comparateur reçoit le signal analogique à convertir (fig. 12.1). Quand la sortie du comparateur est égale 0, c'est-à-dire qu'il y a égalité sur les entrées entre le signal à mesurer, on arrête le comptage des impulsions qui avait débuté quand la tension en dents-de-scie était égale à 0.

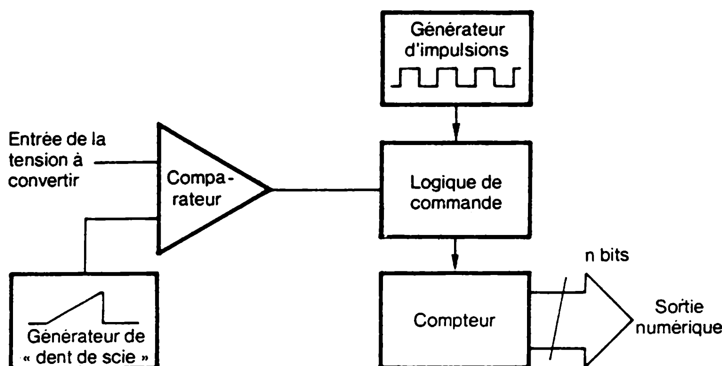


Fig. 12-1. Conversion par simple rampe.

Ce nombre est donc proportionnel à la fréquence du générateur d'impulsions et au temps de comptage, c'est-à-dire au temps qu'a mis la tension en dents-de-scie pour atteindre le signal à convertir. La fréquence du signal en dents-de-scie étant fixe et connue ainsi que celle du générateur d'impulsions, le nombre contenu dans le compteur est un mot binaire proportionnel à la valeur de la tension du signal à mesurer.

Remarque :

Dans ce mode de conversion, il est nécessaire que la fréquence du générateur d'impulsions soit stable, que la variation de la dent-de-scie soit linéaire et que le comparateur de tension ait une grande sensibilité. Toutes ces qualités sont difficiles à réunir, aussi cette méthode manque de précision.

• Conversion par rampe en escalier : (Méthode à propositions de poids).

Le principe consiste à convertir une information numérique provenant d'un compteur, alimenté par un générateur d'impulsions, en une grandeur analogique grâce à un convertisseur numérique-analogique (CNA) et de comparer cette grandeur avec la valeur à mesurer. Quand il y a identité entre ces 2 valeurs, le comptage est interrompu puisque la « bonne valeur » a été trouvée (fig. 12.2).

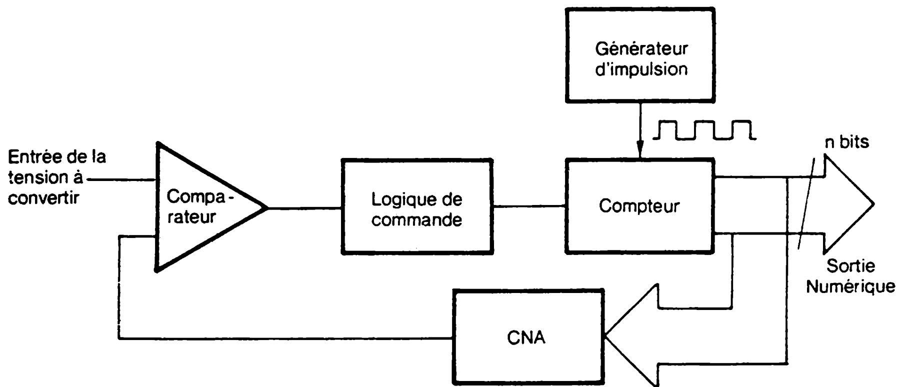


Fig. 12-2. Conversion par proposition de poids.

Remarques :

L'avantage principal par rapport à la méthode précédente est que la comparaison se faisant sur des valeurs progressant par bonds successifs, l'imprécision due au comparateur est annulée.

• Convertisseur tension-fréquence

Le principe consiste à convertir la tension à mesurer en une fréquence proportionnelle et de mémoriser cette fréquence grâce à un compteur qui effectue, entre deux impulsions délivrées par un générateur, le comptage des bits obtenus à la sortie du convertisseur tension-fréquence (fig. 12.3).

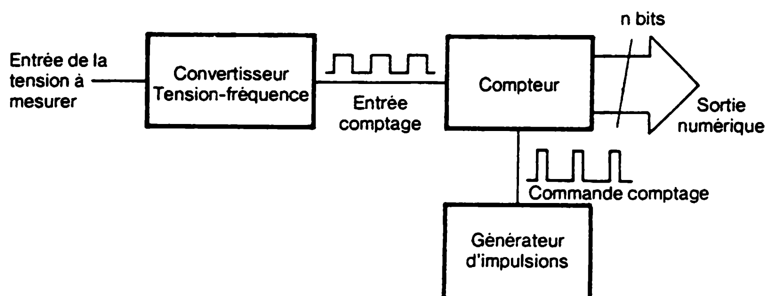


Fig. 12-3. Conversion par la méthode tension-fréquence.

• Conversions par approximations successives

Pour une tension à mesurer donnée, l'opération consiste à déterminer successivement tous les bits du nombre représentatif de cette tension en commençant par le bit de poids le plus fort et en finissant par le bit de poids le plus faible (fig. 12.4).
La détermination du bit de poids le plus fort se fait de la façon suivante :

On compare la tension (Ve) à mesurer avec la moitié d'une tension de référence donnée qui représente la valeur maximale de la tension à mesurer.

Si Ve $\geqslant$ Vréf/2, le bit de poids le plus fort = 1
Si Ve $<$ Vréf/2, le bit de poids le plus fort = 0

On mémorise la valeur de ce bit dans un registre et on passe à la détermination du bit de poids immédiatement inférieur. Pour cela, on retranche à Ve le poids du bit le plus fort soit Vref/2 et on compare le résultat à Vref/4.

Si Ve - Vref/2 $\geqslant \frac{Vref}{4}$ alors le bit vaut 1

Si Ve - Vref/2 $< \frac{Vref}{4}$ le bit vaut 0

Après mémorisation, on continue en retranchant à Ve les poids du bit le plus fort (Vref/2) et du bit précédent (Vref/4) et on compare avec Vref/8 et ainsi de suite jusqu'au bit de poids le plus faible.

Nota :

Le bit de poids le plus fort s'appelle le MSB (Most signifiant bit) et le bit de poids le plus faible, le LSB (Last signifiant bit).

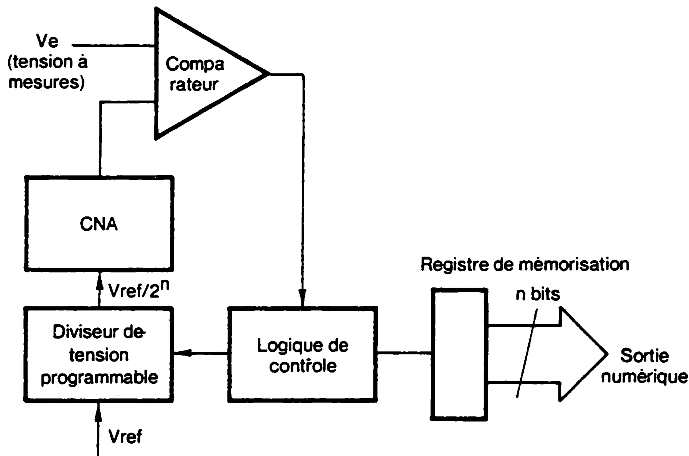


Fig. 12-4. Conversion par approximations successives.

Remarque :

De réalisation complexe, ces convertisseurs présentent l'avantage d'avoir un temps de conversion fixe indépendant de la valeur de l'information analogique à convertir ce qui se prête bien à l'acquisition de données pour un traitement informatique.

• Conversion à intégration par double rampe

Le principe est d'obtenir un temps, pendant lequel on effectue un comptage, proportionnel à la tension à mesurer (fig. 12.5).

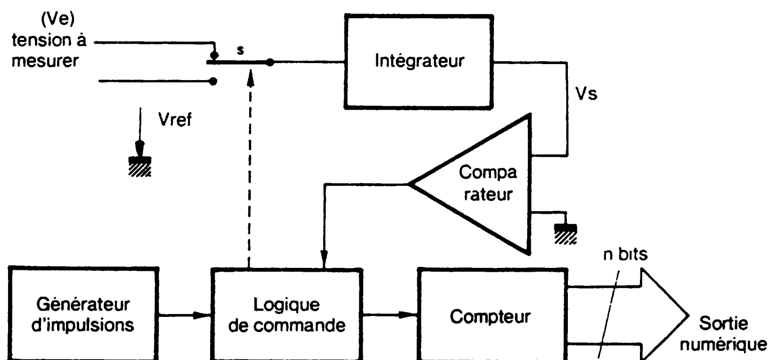
Pendant un temps T_1 (de t_0 à t_1), on applique la tension Ve à mesurer à l'intégrateur qui délivre en sortie une tension.
$Vs = k \cdot Ve \cdot T_1$

Au bout du temps T_1 (à l'instant t_1), la logique de commande fait basculer l'inverseur s qui permet d'appliquer à l'intégrateur, la tension – Vréf.

La sortie Vs de l'intégrateur décroit donc et atteindra la valeur 0 (détectée par le comparateur) à l'instant t_2. On peut donc écrire (en t_1).

$Vs = k \ Ve \ T_1 = k \ Vref \ T_2$

et $T_2 = \frac{T_1}{Vref}$ Ve et comme T_1 est fixé par la logique de commande et que Vréf est connue, il suffit de compter pendant le temps T_2 le nombre d'impulsions provenant du générateur pour obtenir un mot binaire proportionnel à Ve.



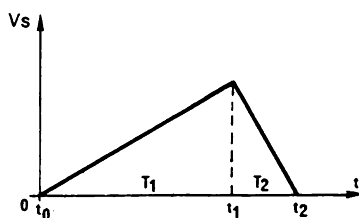


Fig. 12-5. Conversion par intégration à double rampe.

3. Convertisseur numérique-analogique (C.N.A.)

• Principe

On applique à l'entrée du convertisseur un mot binaire de n bits et on veut obtenir en échange un niveau de tension proportionnel à ce mot. Ainsi, si le mot appliqué s'écrit : $a_{n-1} \dots a_0$ ($a_i = 0$ ou 1), la sortie analogique pourra s'écrire :
$Vs = K\,(a_{n-1}/2 + a_{n-2}/4 + \dots + a_0/2^n)$
k étant la quantité analogique mesurable à pleine échelle.

• Le CNA parallèle (fig. 12.6)

Chaque « interrupteur statique » est lié à un bit de l'information numérique qui le commande :

- Si le bit correspondant a_i est égal à 0, cela provoque une connexion à la masse.
- Si le bit correspondant a_i est égal à 1, c'est une connexion à Vréf qui est faite.

Chaque résistance est en progression géométrique de raison 2. On associe au bit de poids le plus fort la plus petite résistance (2 R) et au bit de poids le plus faible, la plus grande (2^n R).
On obtient ainsi un étage sommateur qui délivre en sortie une tension V_s qui peut s'écrire :

$$V_s = Vref\,(\frac{a_{n-1}}{2} + \frac{a_{n-2}}{1} + \dots + \frac{a_0}{2^n})$$

Vref représentant la tension maximale mesurable à un quantum près.

Puisque $V_{s\,MAX} = Vref\,(1 - \frac{1}{2^n}) = Vref - \frac{Vref}{2^n}$

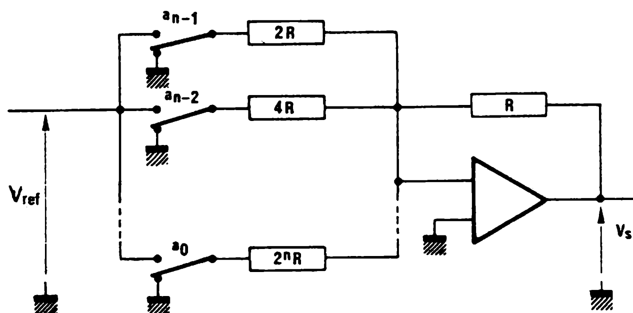


Fig. 12-6. CNA parallèle.

$\frac{Vref}{2^n}$ est la plus petite valeur analogique et correspond au bit de poids le plus faible (LSB).

Remarques

Ce circuit relativement simple est toutefois difficile à réaliser (réseau de résistances en progression géométrique).
On lui préfère le CNA en échelle.

• Le CNA en échelle (fig. 12-7)

En disposant autrement les résistances, on peut obtenir les mêmes résultats que précédemment mais à l'aide de deux types seulement de résistances : R et 2 R.

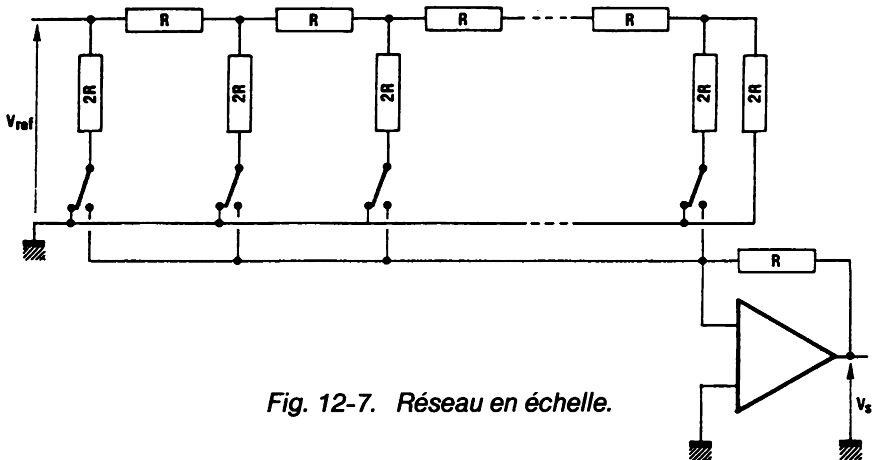


Fig. 12-7. Réseau en échelle.

Remarque

La plupart des CNA actuels utilisent cette méthode de conversion.

4. Exemples de circuit

- ADC 0800 de National Semi Conductor Convertisseur A/N 8 bits.
- DAC 0800 de National Semi Conductor Convertisseur N/A 8 bits.

5. Quelques définitions sur les convertisseurs

Lorsqu'on choisit un circuit chez un constructeur, celui-ci nous donne quelques caractéristiques que nous allons définir aussi bien pour les CAN que pour les CNA.

• Convertisseur analogique-numérique

On s'intéresse surtout à la précision, à la résolution et à la linéarité du circuit.

- **La précision :** on peut distinguer la précision absolue et la précision relative. Cette grandeur s'exprime en fonction du LSB ou en % de la pleine échelle de mesure. De toute façon, il s'agit de l'écart existant entre la tension réelle qu'il faut appliquer et la valeur théorique correspondant au LSB.
- **La résolution :** c'est la différence de tension la plus petite qui appliquée à l'entrée provoque un changement d'état du LSB.
- **Linéarité :** théoriquement, la courbe de transfert d'un CAN est une droite, en réalité on note des écarts entre la courbe de transfert réelle et la droite théorique.
  Les allures des courbes réelles varient suivant les méthodes de conversion employées.
- **Temps de conversion :** c'est le temps s'écoulant entre l'instant où on applique la grandeur analogique à mesurer et le moment où le résultat de la conversion peut être lu. Ce temps peut être variable suivant les méthodes de conversion ou fixe (méthode par approximations successives).

• Convertisseur numérique-analogique

Dans les CNA, on s'intéresse aussi aux caractéristiques précédentes ramenées aux entrées et sorties de ces convertisseurs mais en plus on a l'amplitude de la tension de sortie qui est l'écart de tension obtenu lorsque tous les bits du CNA passent de l'état 0 à l'état 1.

XIII - Visualisation

1. Introduction

On regroupe en général dans les systèmes de visualisation tous les circuits électro-optiques qui transforment une partie de l'énergie électrique qu'ils reçoivent en énergie lumineuse.

2. Systèmes de visualisation

- Tout ou rien (voyant, barrettes lumineuses, etc.).
- Alphanumérique (afficheurs à segments, par matrice de points, par sélection, etc.).

3. Systèmes de visualisation « tout ou rien »

• Technologie de la diode électro-luminescente (LED : Light Emitting Diode)

Principe :

Toute diode (jonction PN) polarisée en direct émet un rayonnement lumineux dû à l'énergie émise par les recombinaisons trou-électron.

Suivant les matériaux constituant la diode, le rayonnement se fait dans une longueur d'onde différente donc avec des couleurs différentes.

Exemples

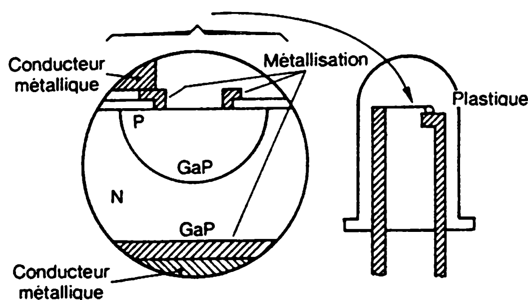
Matériaux	Longueur d'onde	Couleur
(Ge) Germanium	1,88 µm	Infrarouge
(Ga.Al.As) Gallium Aluminium - Arsenic	0,68 µm	Rouge
(Ga.P) Gallium Phosphore	0,54 µm	Vert

Remarque :

La dernière née est réalisée en nitrure de Gallium (GaN) qui : pour une tension directe de 2,4 V devient orange puis à 4 V devient « violette ».

Constitution :

Exemple : LED verte au phosphore de Gallium (fig. 13.1).



• Différentes présentations

Standard : (boîtier SOD63) (fig. 13.2)

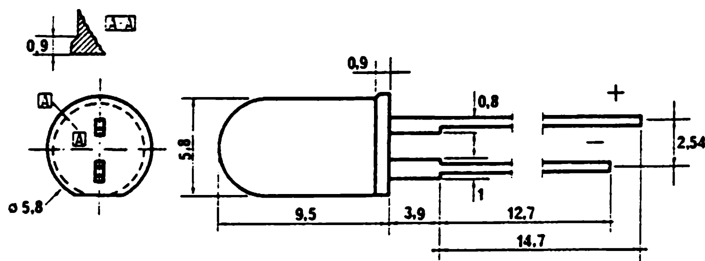


Fig. 13-2. Boîtier standard (LED).

Miniature : (boîtier FO - 78C) (fig. 13.3)

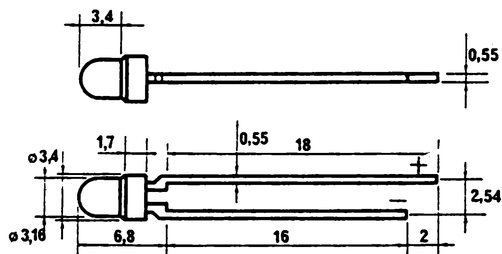


Fig. 13-3. Boîtier miniature (LED).

Subminiature : (boîtier FO - 105) (fig. 13.4)

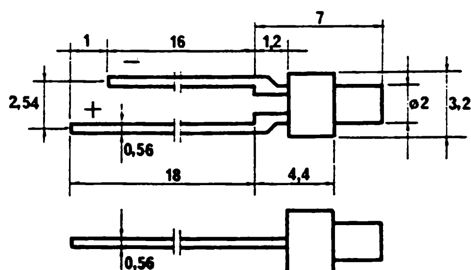


Fig. 13-4. Boîtier subminiature (LED).

Il existe aussi des présentations particulières comme les LEDS rectangulaires, les LEDS en boîtier hermétique, les LEDS à résistance intégrée (fig. 13.5).

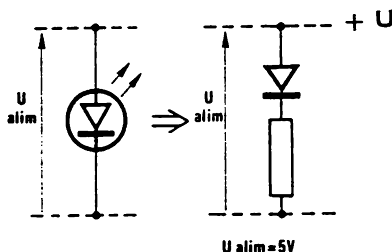
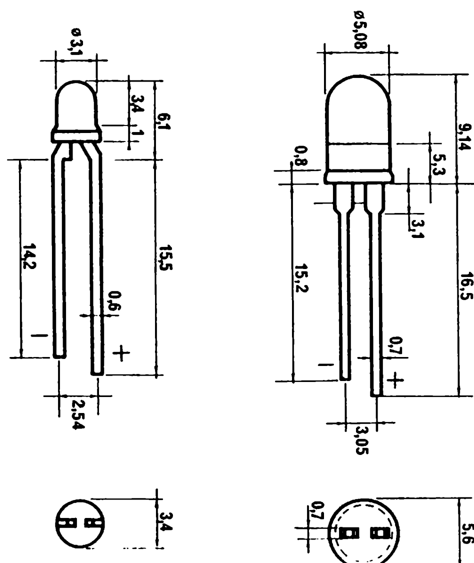


Fig. 13-5. Diode électro-luminescente à résistance intégrée.

Exemple de circuit :
Les 5082-4860 et 5082-4468 de Hewlett Packard.



Il existe aussi des barrettes lumineuses composées de plusieurs diodes électro-luminescentes.

Exemple de circuit :

RTC 901 - RTC 903
RTC 902 - RTC 904

• Utilisation de la diode électro-luminescente

Comme toute jonction PN, la caractéristique directe Courant-Tension présente un seuil dépendant des matériaux constituants la diode. Cette tension du seuil étant en général inférieure à la tension d'alimentation du **voyant,** il faudra ajouter une résistance en série. Il faut donc tenir compte de ces tensions pour le calcul de cette résistance de protection (si celle-ci n'est pas intégrée comme dans la plupart des cas). D'autre part, pour obtenir une luminosité suffisante, on choisira le courant direct nécessaire (en général 10 - 15 mA : donné par le constructeur).

Exemple de calcul de la résistance de protection

Tension d'alimentation : 5 V.
Diode choisie : CQX51 de RTC.
Caractéristique courant-tension : $I_F = f$ (VF) (fig. 13.6).

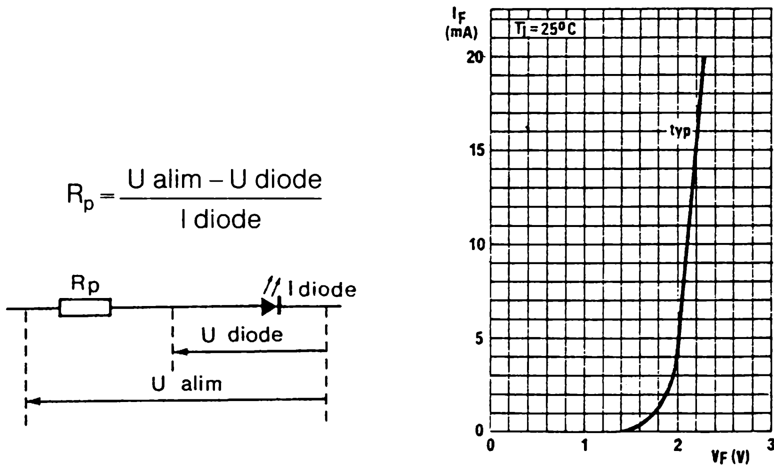


Fig. 13-6. Caractéristique IF = f (VF).

D'après les caractéristiques I_F nominal = 10 mA le graphe I_F = f (VF) nous permet de déterminer U diode (VF) celle-ci est de : 2,1 V
D'où :

$$R_p = \frac{5 - 2{,}1}{10.10^{-3}} = 290\ \Omega$$

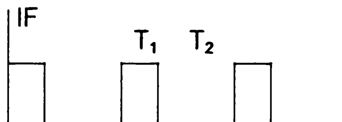
Remarque

Il existe une autre façon d'alimenter la diode électro-luminescente sans utiliser de résistance de protection. On alimente celle-ci avec un signal carré dont la valeur moyenne ne dépasse pas V_F nominal. Cette méthode est beaucoup utilisée dans les systèmes programmés (micro-systèmes utilisant de l'affichage à LED) : les LEDS étant alimentées séquentiellement par programme, elles reçoivent donc des impulsions de courant qui donnent une valeur moyenne nécessaire pour la luminosité demandée.

Exemple

Calculer $\eta = \frac{T_1}{T_1 + T_2}$ % pour obtenir

un If moy de 10 mA si IFMAX = 180 mA



IFMAX = 180 mA

$IFmoy = IFmax \times \frac{T_1}{T_1 + T_2}$ d'où

$\frac{T_1}{T_1 + T_2} = \frac{Ifmoy}{Ifmax}$ et $\eta\% = \frac{Ifmoy}{IFmax} \times 100$

A° N° : $\eta\,\% = \frac{10}{180} = 100 = 5{,}5\ \%$

4. Systèmes de visualisation alphanumérique

• Afficheurs à segments

Ils sont obtenus par association dans un même boîtier, d'un certain nombre de segments permettant la représentation de chiffres ou de lettres.



Configurations :

– Afficheur 7 segments (le plus utilisé)

Il permet la représentation des chiffres 0 à 9, plus éventuellement, les 6 symboles supplémentaires hexadécimaux (fig. 13.7).

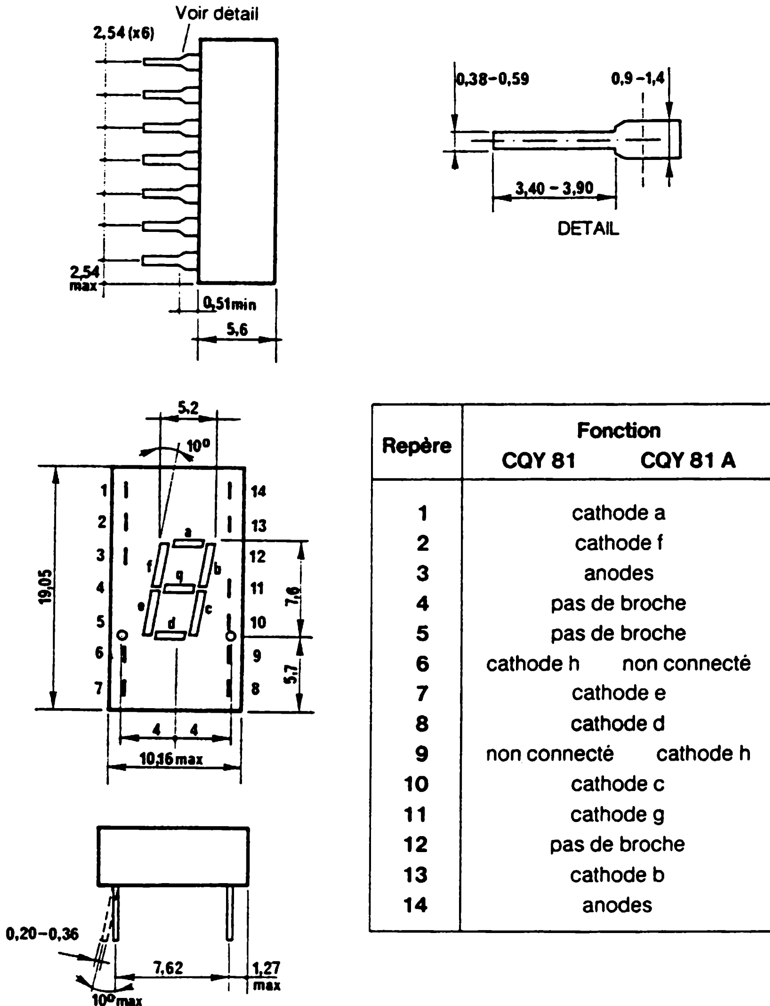
0 123456789 AbCDEF

Fig. 13-7. Représentation des nombres 0 à F avec un afficheur 7 segments.

Exemples de circuits

CQY 84 de RTC

CQY et CQY 81 A de RTC (fig. 13.8)



Repère	Fonction CQY 81	Fonction CQY 81 A
1	cathode a	
2	cathode f	
3	anodes	
4	pas de broche	
5	pas de broche	
6	cathode h	non connecté
7	cathode e	
8	cathode d	
9	non connecté	cathode h
10	cathode c	
11	cathode g	
12	pas de broche	
13	cathode b	
14	anodes	

Fig. 13-8. Brochage des circuits CQY 81 et CQY 81 A.

- Afficheur 9 segments 

  L'afficheur 9 segments permet en plus de visualiser les symboles arithmétiques + et –
- Afficheurs alphanumériques 14 segments (fig. 13.9).



Fig. 13-9. Afficheur 14 segments.

- ou 16 segments (fig. 13.10).



Fig. 13-10. Afficheur 16 segments.

Ils permettent la représentation des chiffres et des lettres.

Remarque :

Certains afficheurs alphanumériques possèdent en plus, des segments permettant de visualiser par exemple le point décimal ou une virgule.

Exemple
HDSP - 6504 de Hewlett Packard (fig. 13.11).

Possibilités d'affichage (64 caractères ASCII + 18 caractères mathématiques).

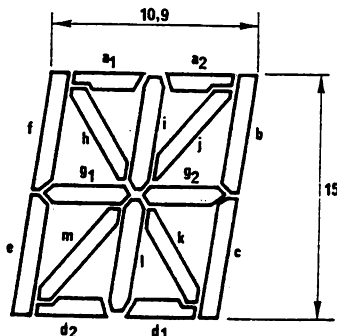


Fig. 13-11. Afficheur 16 segments HD SP - 6504 de HP.

Technologies

Les segments composant les afficheurs sont réalisés avec des diodes électro-luminescentes à anode commune ou à cathode commune (fig. 13.2) ou des cristaux liquides.

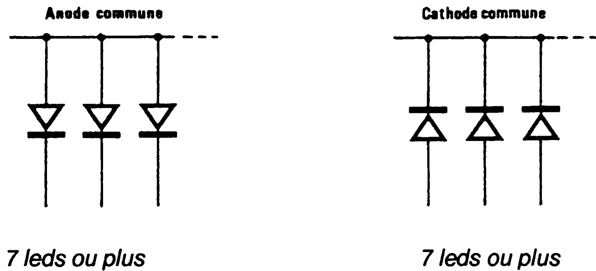


Fig. 13-12. Anode ou cathode commune.

Cristaux liquides : (Principes)

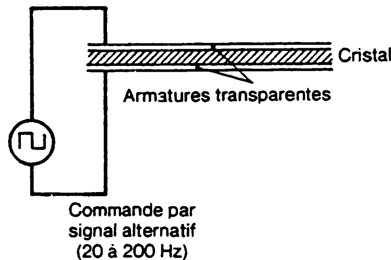


Fig. 13-13. Principe cristaux liquides.

Des armatures transparentes forment les segments vus précédemment (fig. 13.13). Le cristal qui se trouve entre les deux armatures réfléchit ou non la lumière ambiante s'il se trouve excité par un champ électrique convenable.

On peut retenir 3 principes :

– La diffusion dynamique :

On applique au cristal liquide homogène une tension supérieure à une tension de seuil à partir de laquelle se produit des distorsions dans le cristal qui n'est plus transparent et diffuse alors la lumière incidente.

Le contraste entre la zone transparente et la zone diffuse est alors visible. Les premiers afficheurs à cristaux liquides étaient réalisés suivant ce principe.

– Nématique en hélice :

Les afficheurs actuels utilisent ce principe, qui est basé sur la polarisation de la lumière due à des forces diélectriques provoquées par le champ électrique.
L'avantage principal est que la consommation est très faible mais le contraste obtenu dépend de l'angle de vision de l'afficheur.

– Dichroïque :

C'est un des procédés les plus récents. Il est basé sur la polarisation et la réfraction de la lumière et permet suivant la tension appliquée d'obtenir des segments de couleurs différentes.

• Afficheurs à matrice de points

Chaque point est constitué d'une diode électro-luminescente et la sélection de certaines de ces diodes parmi les autres permettent la représentation de caractères alphanumériques.

Configurations :

Mêmes possibilités que les afficheurs 7 segments (de moins en moins utilisés) (fig. 13.14).

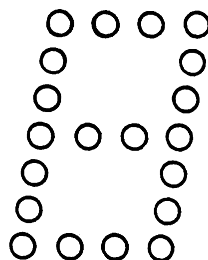


Fig. 13-14. Afficheur 7 segments de points

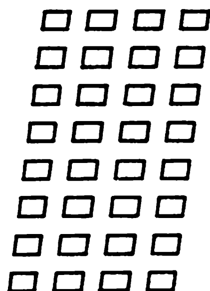
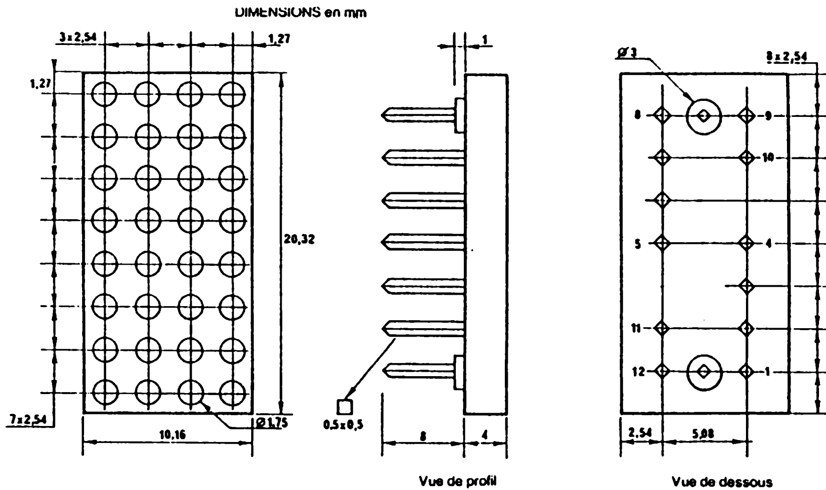


Fig. 13-15. Afficheur à matrice 32 points (4 x 8).

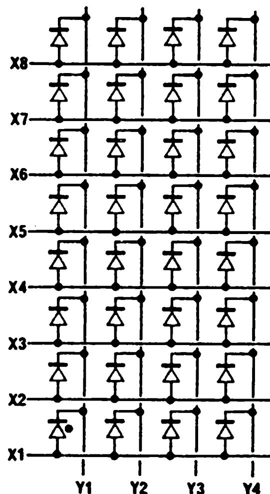
Matrice 32 points (4 × 8).
Permet la représentation des caractères alphanumériques (fig. 13.15).

Exemple de circuits

CQW 32 V et CQW 32 J de RTC.
Brochage de l'afficheur CQW 32 de RTC (fig. 13.16).



DONNÉES MÉCANIQUES



Lignes		Colonnes	
1	X1	9	Y1
2	X2		
3	X3	10	Y2
4	X4		
5	X5	11	Y3
6	X6		
7	X7	12	Y4
8	X8		

Fig. 13-16. Brochage et schéma de l'afficheur CQW 32.

Matrice 35 points (5 × 7).
C'est la configuration la plus utilisée actuellement. L'association des LEDS en ligne colonne suit le même principe que le circuit vu précédemment (fig. 13.17).

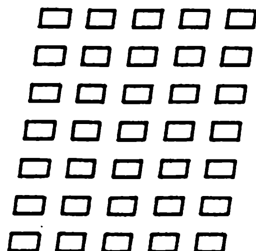


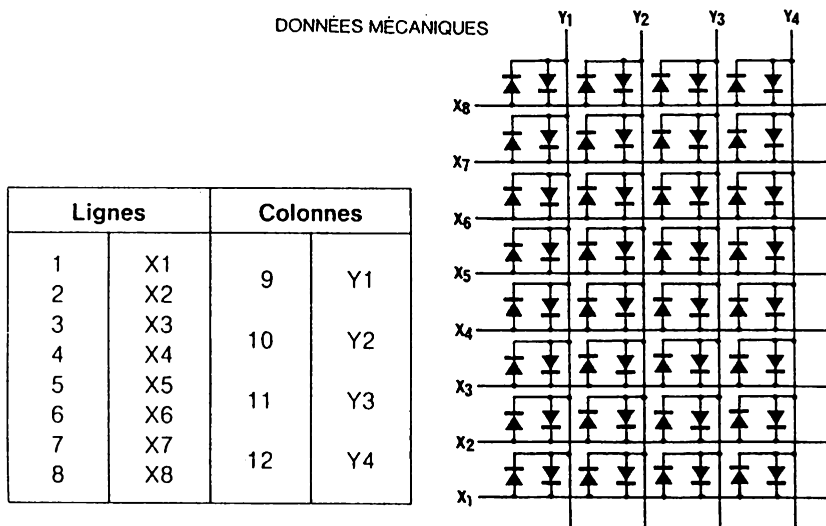
Fig. 13-17. Afficheur à matrice 35 points (5 × 7).

Remarque

Certains constructeurs ont réalisé des afficheurs matrice par points dont le point est réalisé par 2 LEDS différents connectées en antiparallèle ce qui permet d'obtenir 3 couleurs.

Exemple

CQT 32 de RTC (fig. 13.18).



Lignes		Colonnes	
1	X1	9	Y1
2	X2		
3	X3	10	Y2
4	X4		
5	X5	11	Y3
6	X6		
7	X7	12	Y4
8	X8		

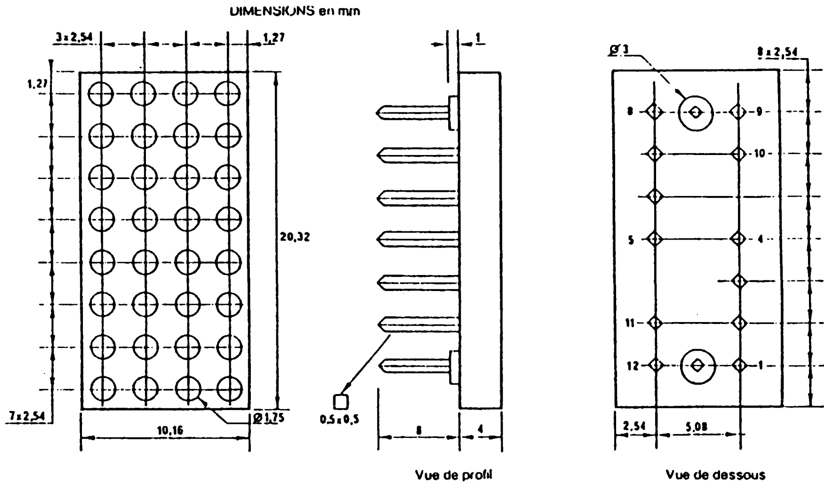


Fig. 13-18. Caractéristiques de l'afficheur CQT 32 de RTC.

5. Systèmes de commande

• Affichage par segments

Le principe est toujours le même, on utilise un transcodeur qui reçoit le code utilisé (exemple : BCD) et qui alimente chaque segment (exemple 7 segments : fig. 13.19).

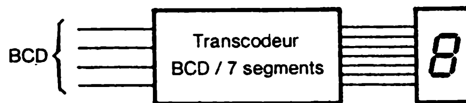


Fig. 13-19. Transcodeur BCD/7 segments pour la commande de l'afficheur.

Il existe de nombreux circuits permettant ce transcodage.
Ils peuvent en outre être équipés d'un registre permettant la mémorisation du dernier code envoyé (pour garder l'affichage).

Exemples

Pour les afficheurs 7 segments à LEDS.

Circuit 4511 (fig. 13.20).

Entrées (Inputs)							Sorties (Outputs)							
$\overline{EL}$	$\overline{I}_B$	$\overline{I}_{LT}$	A_3	A_2	A_1	A_0	a	b	c	d	e	f	g	Affichage (display)
X	X	L	X	X	X	X	H	H	H	H	H	H	H	8
X	L	H	X	X	X	X	L	L	L	L	L	L	L	blanc (blank)
L	H	H	L	L	L	L	H	H	H	H	H	H	L	0
L	H	H	L	L	L	H	L	H	H	L	L	L	L	1
L	H	H	L	L	H	L	H	H	L	H	H	L	H	2
L	H	H	L	L	H	H	H	H	H	H	L	L	H	3
L	H	H	L	H	L	L	L	H	H	L	L	H	H	4
L	H	H	L	H	L	H	H	L	H	H	L	H	H	5
L	H	H	L	H	H	L	L	L	H	H	H	H	H	6
L	H	H	L	H	H	H	H	H	H	L	L	L	L	7
L	H	H	H	L	L	L	H	H	H	H	H	H	H	8
L	H	H	H	L	L	H	H	H	H	L	L	H	H	9
L	H	H	H	L	H	L	L	L	L	L	L	L	L	blanc
L	H	H	H	L	H	H	L	L	L	L	L	L	L	blanc
L	H	H	H	H	L	L	L	L	L	L	L	L	L	blanc
L	H	H	H	H	L	H	L	L	L	L	L	L	L	blanc
L	H	H	H	H	H	L	L	L	L	L	L	L	L	blanc
L	H	H	H	H	H	H	L	L	L	L	L	L	L	blanc
H	H	H	X	X	X	X				*				*

H = HIGH state (the more positive voltage) État HAUT (tension la plus positive)
L = LOW state (the less positive voltage) État BAS (tension la moins positive)
W = state is immaterial État indifférent.

*Depend du code BCD appliqué pendant le front descendant de $\overline{EL}$.

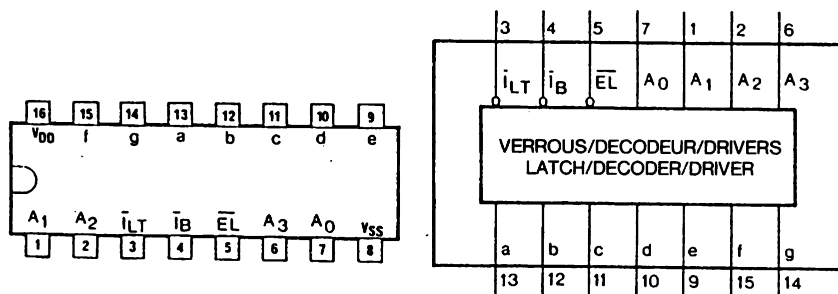


Fig. 13-20. Brochage et table de vérité du circuit 4511.

Pour les afficheurs 7 segments à cristaux liquides.

Circuit 4543 (fig. 13.21)

Ce circuit pratiquement identique au précédent est prévu pour la commande des afficheurs 7 segments à cristaux liquides.

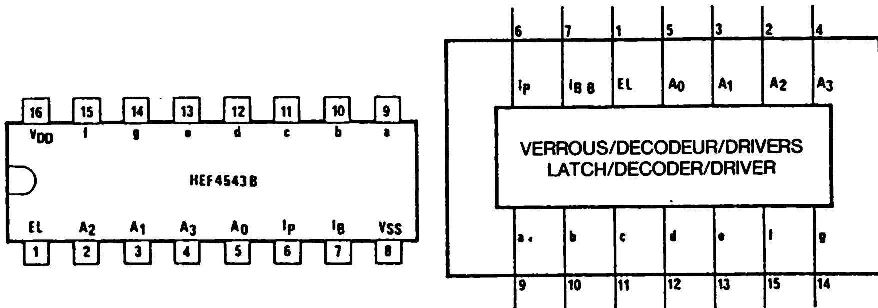


Fig. 13-21. Brochage du circuit 4543.

Description :

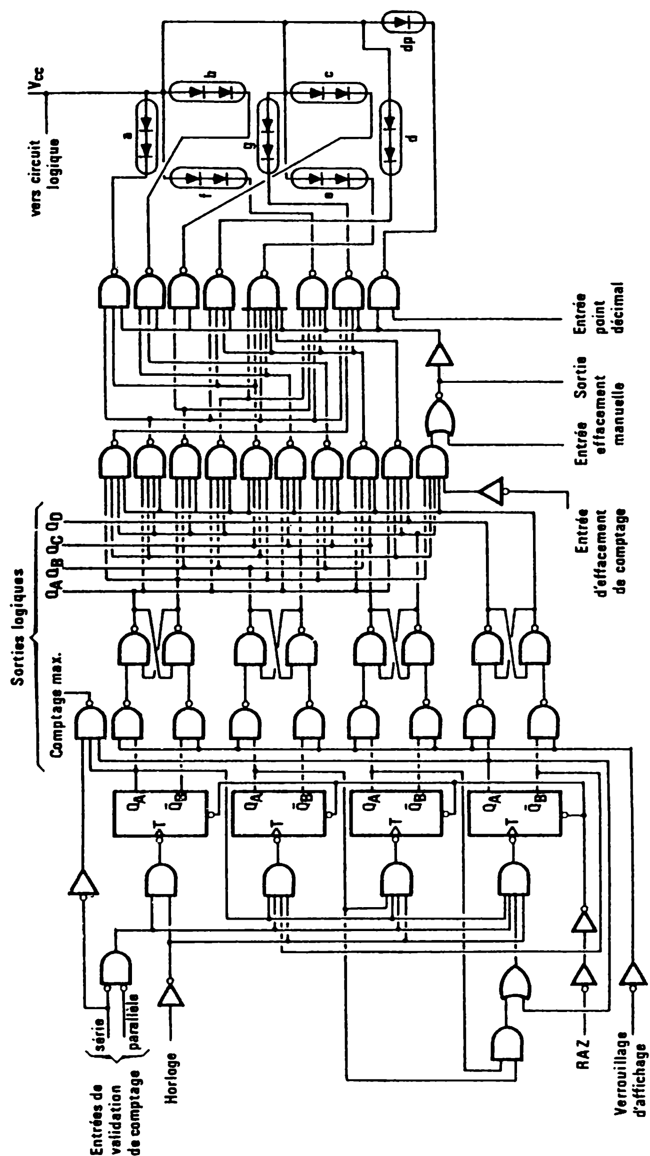
Les entrées BCD sont A_0 à A_3 les sorties 7 segments (a à g).
EL est l'entrée d'inhibition du registre interne (actif au niveau haut).
IB est l'entrée permettant l'extinction de l'afficheur (actif au niveau haut).
Un signal carré doit être appliqué entre l'entrée Ip et le point commun des segments de l'afficheur.
Quand EL passe à 0, la dernière information présente sur les entrées A_0 à A_3 est stockée dans le registre et permet un affichage stable du code BCD correspondant.
Quand EL est à 1, l'affichage dépend de la valeur BCD appliquée sur les entrées A_0 à A_3.

Remarque :

Comme vu précédemment pour les « voyants » à LED, les circuits précédents doivent être suivis de résistances de protection pour chaque segment alimenté (sauf pour les afficheurs à cristaux liquides).

Nota :

Certains constructeurs ont intégré dans un même boîtier, l'afficheur, les résistances de commande, le transcodeur et même un compteur.
C'est le cas des TIL 306 - TIL 307 de Texas Instrument (fig. 13.22).


vers circuit logique
Vcc
a
b
c
d
e
f
g
dp
Sorties logiques
Comptage max.
Q_A Q_B Q_C Q_D
Entrées de validation de comptage
série
parallèle
Horloge
RAZ
Verrouillage d'affichage
Entrée d'effacement de comptage
Entrée effacement manuelle
Sortie effacement manuelle
Entrée point décimal

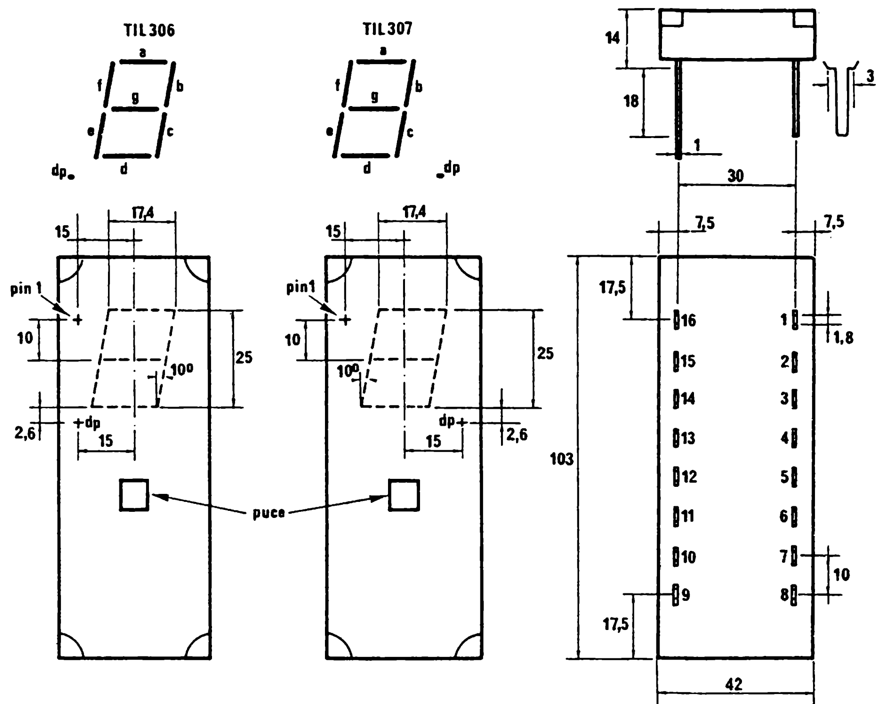


Fig. 13-22. Circuit TIL 306 de TI.

Remarques pour afficheurs 14 ou 16 segments :

Le moyen le plus simple pour réaliser la visualisation des caractères est de réaliser le transcodeur avec une ROM. Elle contiendra un certain nombre de mots de 14 bits (Afficheur 14 segments) ou de 16 bits (Afficheur 16 segments). Ce nombre de mots sera fonction du nombre de caractères à visualiser.

Ainsi, si on veut visualiser sur un afficheur 16 segments par exemple, les lettres majuscules, minuscules ainsi que les 10 chiffres, on peut se contenter d'une ROM de 64 mots de 16 bits.

Cette ROM de 64 mots possèdera donc un bus d'adresses de 6 bits ($2^6 = 64$) et l'adresse appliquée correspondra au caractère à visualiser demandé.

La mémorisation de l'adresse pourra par exemple être faite par un registre de 6 bits (fig. 13.23).

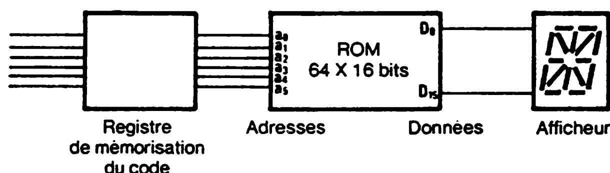


Fig. 13-23. Transcodage pour la visualisation.

• Affichage par points

Le principe est le même que précédemment, le transcodeur est une ROM de n mots (nombre de caractères que l'on veut afficher : en général 64) de 32 bits (Matrice 4 × 8) ou 35 bits (Matrice 5 × 7).

• Commande multiplexée

Pour visualiser N caractères, on pourrait utiliser N systèmes « mémoires - transcodeurs - afficheurs ». Mais si ce nombre est grand, l'ensemble de ces circuits pourrait très rapidement s'avérer répétitif et le nombre d'interconnexion très élevé. On va donc réaliser un multiplexage de la commande.

Dans cette configuration, les segments de même nom de chaque afficheur sont reliés entre eux. Les données multiplexées apparaissent en même temps sur tous les afficheurs mais chacun est validé à son tour.

Le principe est donc d'adresser séquentiellement les afficheurs à une fréquence telle que chaque afficheur semble alimenté continuellement. En général, à partir d'une fréquence de 100 Hz, on ne voit plus l'extinction du segment ou du point (LED).

D'autre part, les registres de mémorisation vont être remplacés par une mémoire qui en échange de l'adresse de l'afficheur délivrera le mot nécessaire à l'affichage.

On utilisera, si le nombre de caractères est élevé, un visualisateur multi-afficheur dans lequel chaque segment ou point de même position est relié en un point (l'anode par exemple pour les LEDs) et où la commande se fera au niveau de chaque point (le « commun » cathode de chaque afficheur par exemple pour les LEDs).

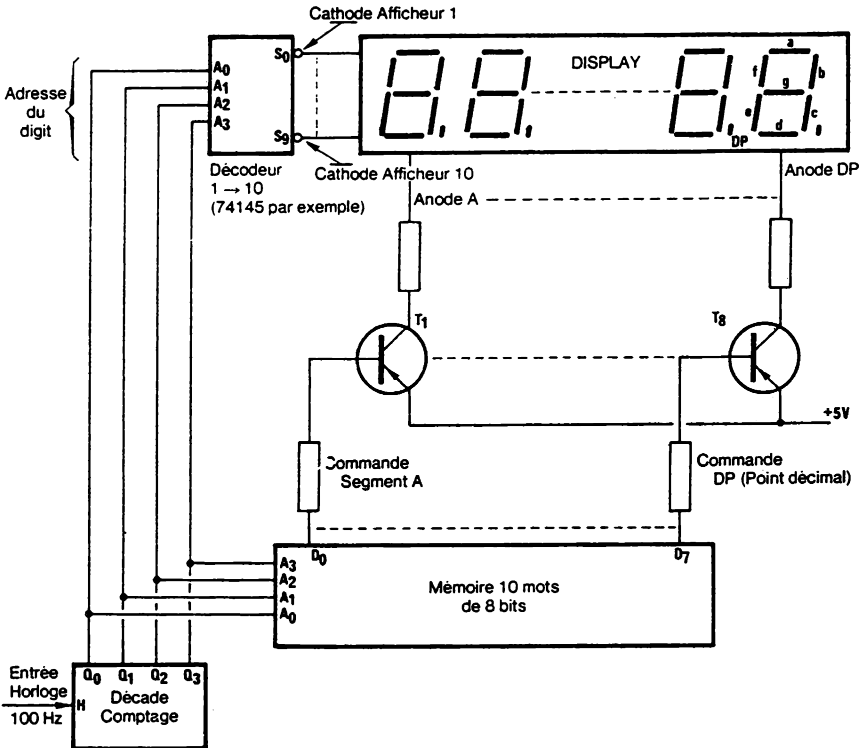


Fig. 13-24. Commande multiplexée d'afficheur.

Exemple

Système de commande d'un « display » 10 digits d'une calculatrice (Principe) (fig. 13.24).

Si la mémoire contient le contenu de chaque afficheur, il suffit d'adresser séquentiellement à une fréquence de 100 Hz les afficheurs pour qu'ils « paraissent » tous allumés.

XIV - Synthèse

1. Exemple : cartes d'entrées analogiques pour micro-ordinateurs.

On se propose d'étudier une carte d'entrées analogiques permettant l'acquisition et le traitement de grandeurs analogiques dans un ordinateur (manettes de jeux, souris, etc.).

2. Données du problème

On veut réaliser une carte dotée de 16 entrées analogiques permettant le traitement numérique (8 bits) de celle-ci.

3. Synoptique (fig. 14.1)

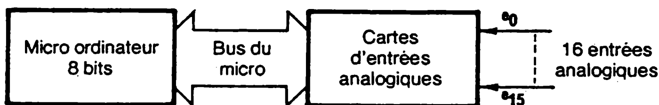


Fig. 14-1. Micro-ordinateur avec entrées analogiques.

4. Principe

La carte d'entrées analogiques sera composée d'un convertisseur analogique/numérique à 16 entrées multiplexées. La sortie numérique (8 bits) du convertisseur alimentera le bus de données du micro-ordinateur. Un décodage du bus d'adresses (16 bits) sera nécessaire pour la sélection du convertisseur.

5. Décodage adresse

On désire que les 16 entrées analogiques se trouvent à des adresses jointives de la pagination mémoire. D'autre part, un pré-découpage de l'espace mémoire est déjà réalisé dans le micro-ordinateur de la façon suivante :
L'espace mémoire étant de 64 K octets (2^{16}), celui-ci est découpé en 16 pages de 4 K chacunes et désignées par P_0 à P_{15}.
Seules les pages P_8 à P_{10} sont disponibles à l'utilisateur pour d'éventuelles extensions comme c'est le cas ici.

Exemple

P_8 correspond aux adresses 8000 à 8 FFF en hexadécimal.
C'est dans cette page (P_8) que l'on trouvera la carte d'entrées analogiques.
Il faut, bien entendu, s'arranger pour que tout l'espace de 4 K octets ne soit pas utilisé par la carte puisque 16 adresses suffisent.
Pour cela, il est nécessaire d'abord de « découper » cette page de 4 K en 16 sous-pages de 256 octets chacunes désignées par S_0 à S_{15}, et ensuite de n'utiliser que 4 adresses parmi 256 à l'intérieur de l'une de ces sous-pages.
On choisira par exemple dans la sous-page S_0 (adresses comprises entre 8000 et 80 FF) les 16 premières adresses (adresses comprises entre 8000 et 800F).

6. Schéma de principe proposé (fig. 14.2)

7. Fonctionnement

Lorsque l'adresse émise par le micro-ordinateur est comprise entre 8000 et 800F en hexadécimal, la sortie P_8 est active et sélectionne le décodeur binaire à 4 entrées, 16 sorties (1 sortie active parmi 16).

Puisque $A_8 = A_9 = A_{10} = A_{11} = 0$ (bits A_8 à A_{11} du bus d'adresse) S_0 est active et sélectionne le comparateur binaire de 2 mots de 4 bits.

Ce dernier aura sa sortie active quand les mots appliqués seront égaux.

C'est ici le cas puisque $A_4 = A_5 = A_6 = A_7 = 0$ (bits A_4 à A_7 du bus d'adresse) et que les 4 bits de l'autre entrée sont à 0 (4 fils reliés à la masse).

Le convertisseur A/N sera donc sélectionné et l'une des 16 combinaisons (2^4) appliquée sur celui-ci par les 4 bits d'adresse A_0 à A_3 permettra la sélection de l'une des entrées (e_0 à e_{15}).
Le résultat de la conversion se trouvera appliquée sur le bus de données (8 fils) ; si on choisit un convertisseur 8 bits ; pour y être traité par le micro-ordinateur.

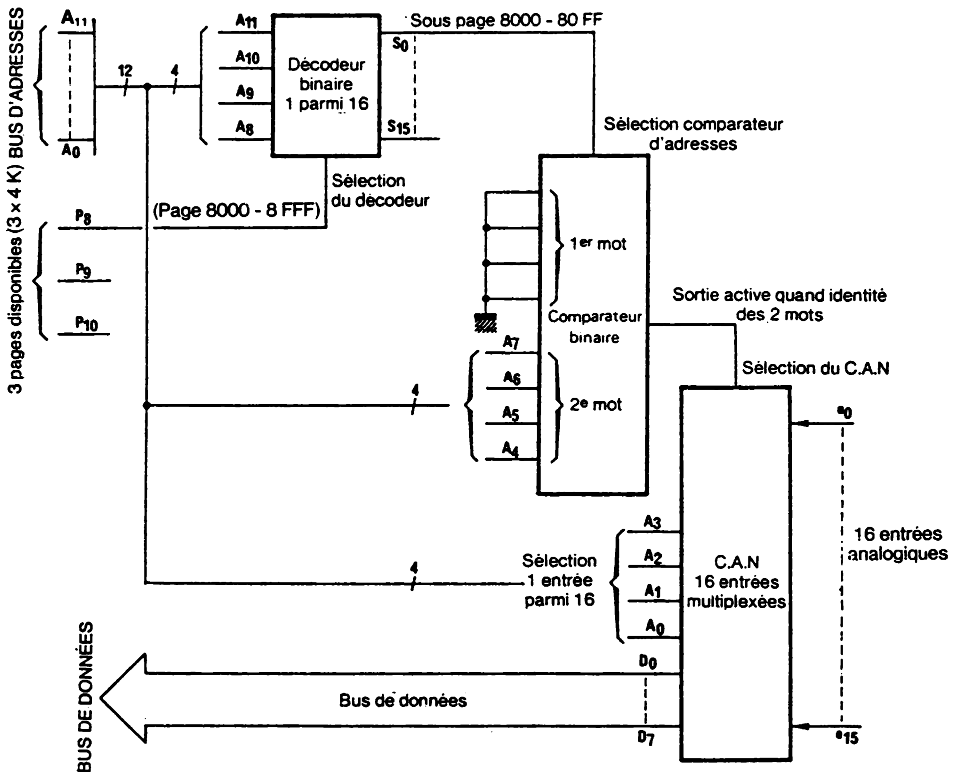


Fig. 14-2. Principe de la carte d'entrées analogiques.

Exemples de circuits choisis pour la réalisation :

- **Convertisseur analogique - numérique** 16 entrées multiplexées : ADC 0816 de National Semiconductor.
- **Décodeur binaire** 1 parmi 16 : 74 LS 154 (technologie TTL, LS).
- **Comparateur binaire :** 74 LS 85 (TTL, LS).

Annexe 1

Table alphanumérique des circuits intégrés principaux TTL (série 74) d'après le Guide de l'ingénieur de RTC, avec l'aimable autorisation de la Radiotechnique Compélec.

Nota :

A l'exception de certains produits particuliers, les numéros de type sont identiques chez tous les constructeurs de circuits intégrés TTL comme MOTOROLA, TEXAS INSTRUMENTS, NATIONAL SEMICONDUCTOR, etc.

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
00	quadruple porte ET-NON à 2 entrées	•	•	•	•
02	quadruple porte OU-NON à 2 entrées	•	•	•	•
03	quadruple porte ET-NON à 2 entrées, collecteur ouvert	•		•	
04	sextuple inverseur	•	•	•	•
05	sextuple inverseur, collecteur ouvert	•	•	•	
06	sextuple inverseur de puissance, collecteur ouvert	•			
07	sextuple porte de puissance, collecteur ouvert	•			

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
08	quadruple porte ET à 2 entrées	•	•	•	•
09	triple porte ET à 2 entrées, collecteur ouvert		•		
10	quadruple porte ET-NON à 3 entrées	•	•	•	•
11	triple porte ET à 3 entrées	•	•	•	•
13	double trigger à 4 entrées	•	•		•
14	sextuple trigger	•	•		•
16	sextuple inverseur de puissance, collecteur ouvert	•			
17	sextuple porte de puissance, collecteur ouvert	•			
20	double porte ET-NON à 4 entrées	•	•	•	•
21	double porte ET à 4 entrées		•		
26	quadruple porte ET-NON à 2 entrées, collecteur ouvert	•	•		
27	triple porte OU-NON à 3 entrées	•			
30	porte ET-NON à 8 entrées	•	•		
32	quadruple porte OU à 2 entrées	•	•	•	•
33	quadruple porte de puissance OU-NON à 2 entrées	•	•		
37	quadruple porte ET-NON de puissance à 2 entrées	•	•	•	•
38	quadruple porte ET-NON de puissance à 2 entrées collecteur ouvert	•	•	•	•
40	double porte ET-NON de puissance à 4 entrées	•	•	•	•
45	décodeur BCD/décimal sortie collecteur ouvert, haute tension	•			
50	double porte ET-OU-NON à 2 fois 2 entrées + expanseur	•			
51	double porte ET-OU-NON à 2 fois 2 entrées	•	•	•	

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
64	porte ET-OU-NON à 4.2.3.2. entrées			•	•
73	double bascule JK-ET maître esclave	•	•		
74	double bascule D	•		•	•
74A	double bascule D		•		
75	quadruple bascule D	•	•		
76	double bascule JK maître esclave avec RAZ	•	•		
83	additionneur 4 bits avec retenue anticipée	•			
83A	additionneur 4 bits avec retenue anticipée		•		
85	comparateur 4 bits	•	•	•	•
86	quadruple OU exclusif à 2 entrées	•	•	•	•
90	décade asynchrone	•	•		
91	registre à décalage 8 bits	•			
92	diviseur par 12		•		
93	compteur 4 bits asynchrone		•		
94	registre à décalage 4 bits (entrée parallèle, sortie série)	•			
95	registre à décalage 4 bits	•			
95B	registre à décalage 4 bits droite/gauche		•		
96	registre à décalage 5 bits		•		
107	double bascule JK maître esclave	•	•		
109	double bascule JK à déclenchement sur front montant	•	•		•
112	double bascule JK à déclenchement sur front descendant		•	•	•
113	double bascule JK à déclenchement sur front montant		•	•	•
114	double bascule JK				•
116	double mémoire 4 bits avec RAZ	•			

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
121	monostable	•			
123	double monostable redéclenchable avec RAZ	•			
125	quadruple porte de puissance avec sortie 3 états	•	•		
126	quadruple porte de puissance avec sortie 3 états		•		
128	quadruple porte OU-NON de puissance à 2 entrées	•			
132	quadruple trigger de Schmitt	•	•		•
133	porte ET-NON à 13 entrées			•	
134	porte ET-NON à 12 entrées et sortie 3 états			•	
136	quadruple OU exclusif, collecteur ouvert		•		
138	décodeur-démultiplexeur		•	•	•
139	décodeur-démultiplexeur		•	•	•
140	double émetteur de lignes ET-NON à 4 entrées			•	
145	décodeur BCD, décimal avec sortie collecteur ouvert, haute tension	•			
148	codeur de priorité à 8 entrées	•			•
150	multiplexeur 16 → 1	•			
151	multiplexeur 8 → 1	•	•	•	•
153	double multiplexeur 4 → 1	•	•	•	•
154	décodeur/démultiplexeur 4 → 16	•	•		
155	double décodeur/démultiplexeur 2 → 4	•	•		
156	décodeur/démultiplexeur 2 → 4		•		
157	quadruple démultiplexeur 2 → 1	•	•	•	•
158	quadruple démultiplexeur	•	•	•	•
160	décade synchrone 4 bits	•			•

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
161	compteur binaire synchrone 4 bits	•			•
161A	compteur binaire synchrone 4 bits		•		
162A	décade synchrone 4 bits		•		•
163	compteur binaire synchrone 4 bits				•
163A	compteur binaire synchrone 4 bits		•		
164	registre à décalage 8 bits, sorties parallèles	•	•		•
165	registre à décalage 8 bits, entrées parallèles	•			
166	registre à décalage 8 bits	•			
168	compteur décimal, déclenchement front montant et descendant		•	•	•
169	compteur binaire, déclenchement front montant et descendant		•	•	•
170	mémoire 4 mots de 4 bits	•	•		
172	double mémoire 8 bits			•	
174	sextuple bascule D avec RAZ	•	•	•	
175	quadruple bascule D à déclenchement sur front montant	•	•	•	•
181	unité arithmétique et logique 4 bits	•	•	•	•
182	générateur de retenue anticipée			•	•
190	compteur-décompteur	•			•
191	compteur-décompteur 4 bits synchrone	•	•		•
192	compteur-décompteur BCD synchrone avec entrées de prépositionnement		•		•
193	compteur-décompteur 4 bits synchrone	•	•		•
194	registre à décalage universel bidirectionnel 4 bits		•	•	•

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
194A	registre à décalage universel bidirectionnel 4 bits		•		
195	registre à décalage 4 bits, entrées parallèles	•		•	•
195A	registre à décalage 4 bits, entrées parallèles		•		
197	compteur binaire 4 bits à entrées de prépositionnement		•		
221	double monostable avec trigger de Schmitt en entrée	•			
225	FIFO			•	
240	inverseur de puissance à 3 états 8 bits		•	•	•
241	porte de puissance à 3 états 8 bits		•	•	•
242	inverseur de puissance à 3 états 8 bits		•	•	•
243	porte de puissance à 3 états 8 bits		•	•	•
244	octuple porte à 3 états 8 bits		•	•	•
245	octuple porte		•		•
251	multiplexeur 8 → 1 avec sorties 3 états		•	•	•
253	double multiplexeur 4 → 1 avec sorties 3 états		•	•	•
256	double latch adressable 4 bits		•		•
257	quadruple multiplexeur 2 → 1 avec sorties 3 états			•	
257A	quadruple multiplexeur 2 → 1 avec sorties 3 états		•		
258	quadruple multiplexeur 2 → 1 avec sorties 3 états			•	•
258A	quadruple multiplexeur 2 → 1 avec sorties 3 états		•		
259	latch adressable 8 bits		•		•
260	double porte OU-NON à 5 entrées		•	•	

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
266	quadruple porte OU-NON exclusif à 2 entrées		•		
269	compteur/décompteur 8 bits				•
273	octuple latch, 3 états		•		•
279	quadruple bascule RS	•			
280	générateur contrôleur de parité 9 bits			•	•
283	additionneur 4 bits		•		
290	décade asynchrone		•		
293	compteur binaire 4 bits		•		
295B	registre à décalage 4 bits, 3 états		•		
298	quadruple multiplexeur à 2 entrées et mémoire	•	•		
299	octuple registre à décalage/tampon 3 états				•
322	octuple registre à décalage/tampon 3 états				•
323	octuple registre à décalage/tampon 3 états				•
350	registre à décalage 4 bits			•	•
352	double multiplexeur à 4 entrées				•
353	double multiplexeur à 4 entrées				•
365	sextuple porte de puissance à sortie 3 états	•			•
365A	sextuple porte de puissance à sortie 3 états		•		
366	sextuple inverseur de puissance à sortie 3 états	•			•
367	sextuple porte de puissance à sortie 3 états	•			•
367A	sextuple porte de puissance à sortie 3 états		•		
368	sextuple inverseur de puissance à sortie 3 états	•			•

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
368A	sextuple inverseur de puissance à sortie 3 états	•			
373	octuple latch sortie 3 états 8 bits		•	•	•
374	octuple bascule D sortie 3 états 8 bits		•	•	•
375	quadruple latch		•		
377	registre à verrouillage 8 bits sortie 3 états		•		•
378	sextuple bascule D avec validation		•		•
379	quadruple bascule D avec validation				•
390	double décade		•		
393	double compteur binaire		•		
395A	registre à décalage 4 bits, sortie 3 états		•		•
398	bascule 4 bits, sortie vraie et complémentaire				•
399	bascule 4 bits, sortie vraie et complémentaire				•
412	octuple latch multimode				•
431	port de sortie 8 bits				•
432	(1) octuple latch inverseur multimode				•
445	décodeur driver BCD décimal, collecteur multimode		•		
490	double décade à sortie 3 états		•		
521	octuple comparateur				•
524	comparateur 8 bits				•
533	octuple latch inverseur, sortie 3 états				•
534	octuple bascule D inverseur, sortie 3 états			•	•
540	octal driver		•		
541	octal driver		•		
545	octal transceiver			•	•

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
568	compteur/décompteur binaire 4 bits, 3 états		●		●
569	compteur/décompteur binaire 4 bits, 3 états		●		●
579	compteur/décompteur 8 bits				●
588	octal transceiver compatible GPIB				●
595	registre à décalage 8 bits, écriture /latch en sortie				●
596	registre à décalage 8 bits, écriture /latch en sortie				●
597	registre à décalage 8 bits, écriture/latch en sortie				●
598	registre à décalage 8 bits, écriture/latch en sortie				●
604	double latch 8 bits				●
605	double latch 8 bits				●
606	double latch 8 bits				●
607	double latch 8 bits				●
620	octuple transceiver		●		●
621	octuple transceiver		●		●
622	octuple transceiver		●		●
623	octuple transceiver		●		●
630	détecteur/correcteur d'erreur mémoire, 3 états				●
631	détecteur/correcteur d'erreur mémoire, collecteur ouvert				●
640	octuple transceiver		●		
641	octuple transceiver		●		
642	octuple transceiver		●		
645	octuple transceiver		●		
645-1	octuple transceiver bidirectionnel		●		
646	octuple registre et transceiver				●

	Fonction	Standard N 74	LP.Schottky N 74 LS	Schottly N 74 S	FAST N 74 F
647	octuple registre et transceiver				•
648	octuple registre et transceiver				•
649	octuple registre et transceiver				•
670	mémoire 4 mots de 4 bits avec sorties 3 états		•		
673	registre à décalage 16 bits série en entrée/parallèle en sortie				•
674	registre à décalage 16 bits parallèle en entrée/série en sortie				•
675	registre à décalage 16 bits série en entrée/parallèle en sortie				•
676	registre à décalage 16 bits parallèle en entrée/série en sortie				•
779	compteur 8 bits				•

Annexe 2

Table alphanumérique par fonctions des circuits intégrés principaux CMOS (série standard 4000) - d'après le Guide de l'ingénieur RTC, avec l'aimable autorisation de la Radiotechnique Compélec.

Nota :
A l'exception de certains produits particuliers, les numéros de type sont identiques chez tous les constructeurs de circuits intégrés CMOS pour la série standard 4000 comme MOTOROLA, TEXAS INSTRUMENTS, NATIONAL SEMICONDUCTOR, etc.

portes simples

HEF

4000	Double porte OU-NON 3 entrées inverseur
4001 et 4001U	Quadruple porte NON-OU à 2 entrées
4002	Double porte NON-OU à 4 entrées
4011 et 4011 U	Quadruple porte NON-ET à 2 entrées
4012	Double porte NON-ET à 4 entrées
4023	Triple porte NON-ET à 3 entrées
4025	Triple porte NON-OU à 3 entrées
4068	Porte NON-ET à 8 entrées
4071	Quadruple porte OU à 2 entrées
4072	Double porte OU à 4 entrées
4073	Triple porte ET à 3 entrées
4075	Triple porte OU à 3 entrées
4078	Porte NON-OU à 8 entrées
4081	Quadruple porte NON-ET à 2 entrées
4082	Double porte ET à 4 entrées

portes complexes

HEF	**4030**	Quadruple porte OU-exclusif
	4070	Quadruple porte OU-exclusif
	4077	Quadruple porte NON-OU exclusif
	4085	Double porte ET-OU-NON à 2 × 2 entrées
	4086	Porte ET-OU-NON à 4 × 2 entrées

portes de puissance et inverseurs

HEF	**4007U**	Double paire complémentaire inverseur
	4041	Quadruple porte de puissance avec sortie complémentaire
	4049	Sextuple porte de puissance (inverseur)
	4050	Sextuple porte de puissance (non inverseur)
	4069U	Sextuple inverseur
	4502	6 inverseurs sorties 3 états
	40097	Sextuple porte de puissance sortie 3 états (non inverseur)
	40098	Sextuple porte de puissance sortie 3 états (inverseur)

«octals»

HEF	**40240**	Octuple buffer/inverseur
	40244	Octuple portes de puissance, sorties 3 états
	40245	Port bidirectionnel 8 bits, sorties 3 états
	40373	Octuple verrous transparents, sorties 3 états
	40374	Octuple bascule D, sorties 3 états

registres à décalage

HEF	**4006**	Registre à décalage statique 18 étages
	4014	Registre à décalage 8 bits
	4015	Double registre à décalage 4 bits
	4021	Registre à décalage 8 bits
	4031	Registre à décalage 64 bits
	4035	Registre à décalage universel 4 bits
	4094	Registre à décalage et stockage 8 étages pour bus
	4517	Double registre à décalage statique 64 bits
	4557	Registre à décalage 1 à 64 bits de longueur variable
	4731	Quadruple registre à décalage statique 64 bits statique
	40194	Registre à décalage universel bidirectionnel 4 bits
	40195	Registre à décalage universel 4 bits

registres à verrouillage

HEF	**4042**	Quadruple bascule D à verrouillage
	4043	Quadruple bascule NON-OU R/S à verrouillage (sorties 3 états)
	4044	Quadruple bascule NON-ET R/S à verrouillage (sorties 3 états)
	4508	Double verrou 4 bits
	4724	Registre 8 bits adressable à verrouillage

décodeurs, multiplexeurs, démultiplexeurs

HEF 4019 Quadruple multiplexeur à 2 entrées
4028 Décodeur BCD/décimal (1 parmi 10)
4051 Multiplexeur/démultiplexeur analogique à 8 canaux
4052 Double multiplexeur/ démultiplexeur analogique 4 canaux
4053 Triple multiplexeur/démultiplexeur analogique 2 canaux
4067 Multiplexeur 16 → 1
4511 Décodeur/driver 7 segments
4512 Multiplexeur 8 entrées, sorties 3 états
4514 Décodeur/démultiplexeur, 1 parmi 16, avec registre d'entrée (sortie haute)
4515 Décodeur/démultiplexeur, 1 parmi 16, avec registre d'entrée (sortie basse)
4519 Quadruple multiplexeur à 2 entrées
4539 Double multiplexeur à 4 entrées
4543 Décodeur/driver 7 segments cristaux liquides
4555 Double décodeur/ démultiplexeur, 1 parmi 4 (sortie haute)
4556 Double décodeur/ démultiplexeur, 1 parmi 4 (sortie basse)

mémoires

HEF 4505 Mémoire vive 64 × 1 bit
4720 Mémoire vive 256 bits (256 × 1)

fonctions spéciales

HEF 4008 Additionneur 4 bits avec retenue
4016 Quadruple interrupteur bidirectionnel
4046 Boucle à phase asservie (PLL)
4047 Monostable/astable
4066 Quadruple interrupteur bidirectionnel
4093 Quadruple trigger de Schmitt avec entrées ET
4104 Quadruple translateur de tension, sortie 3 états
4527 Multiplicateur BCD
4528 Double monostable
4531 Générateur de parité 13 entrées
4532 Codeur de priorité 8 bits
4538 Double monostable
4541 Circuit d'horloge programmable
4585 Comparateur 4 bits
4738 Interface CEI/IEEE
4750 Synthétiseur de fréquence
4751 Diviseur universel
4752 Contrôle de moteur
4753 Temporisateur
4754 Décodeur bargraph 18 LCD
4755 Circuit de communication de données en série
40106 Sextuple trigger

bascules

HEF 4013 Double bascule D
4027 Double bascule K
4076 Quadruple bascule D sortie 3 états
40174 Sextuple bascule D

compteurs

HEF	**4017**	Compteur Johnson à 5 étages
	4018	Compteur/diviseur par n programmable
	4020	Compteur binaire à 14 étages
	4022	Compteur Johnson à 14 étages, diviseur par 8
	4024	Compteur binaire à 7 étages
	4029	Compteur/décompteur synchrone, binaire/ décimal
	4040	Compteur binaire à 12 étages
	4059	Compteur programmable diviseur par n
	4060	Compteur/décompteur binaire 14 étages (+ oscillateur)
	4510	Compteur/décompteur BCD
	4516	Compteur/décompteur binaire
	4518	Double compteur décimal.
	4520	Double compteur binaire
	4521	Diviseur de fréquence 24 étages
	4522	Compteur à prépositionnement diviseur par N 4 bits (BCD)
	4526	Compteur à prépositionnement diviseur par N 4 bits (binaire)
	4534	Compteur 5 décades en temps réel
	4737	Quadruple décade statique
	40160	Compteur 4 bits décimal RAZ asynchrone
	40161	Compteur 4 bits binaire RAZ asynchrone
	40162	4 bits décimal RAZ asynchrone
	40163	Compteur 4 bits binaire RAZ synchrone
	40192	Compteur/décompteur synchrone 4 bits décimal
	40193	Compteur/décompteur synchrone 4 bits binaire

Bibliographie

- Catalogue *RTC* série *4000 LOCMOS*.

- Catalogue *TEXAS INSTRUMENTS DATA BOOK TTL*.

- Manuel des circuits intégrés analogiques *(TEXAS INSTRUMENTS)*.

- Structure et fonctionnement des ordinateurs de J.P. MEINADIER (Editions *LAROUSSE*).

- De la logique câblée aux microprocesseurs de J.-M. BERNARD et J. HUGON (Editions *EYROLLES)*.

- Circuits intégrés et techniques numériques de R. DELSOL (Editions *CEPADUES)*.

- Mémoires intégrées de H. LILEN (Editions *RADIO)*.

Table des matières

GUIDE PRATIQUE DES SYSTEMES LOGIQUES

D'un accès rapide, cet ouvrage regroupe tous les éléments nécessaires à la compréhension des systèmes logiques et de leurs applications.
Le lecteur y trouvera également, pour chaque fonction et montage s'y rapportant, les éléments de référence comme des exemples de circuits intégrés avec leurs caractéristiques et leur brochage.
C'est un guide pratique pour l'étude, la conception et la réalisation des systèmes logiques.

Principaux chapitres :

— **Les systèmes de numérotation :** binaire, BCD, hexadécimal...
— **Les circuits combinatoires :** décodeur, comparateur, multiplexeur, additionneur...
— **Les circuits séquentiels :** bascules, compteurs, registres...
— **Les mémoires :** ROM, PROM, EPROM, EEROM, RAM... ; plan mémoire, extension mémoire d'un ordinateur...
— **Les convertisseurs analogique-numérique et numérique-analogique :** principes, exemples de circuits, applications...
— **L'affichage :** les afficheurs 8 segments, 16 segments, les circuits de commande des afficheurs...
— **Synthèse :** carte d'entrées analogiques pour micro-ordinateur.



Editions Techniques
et Scientifiques Françaises



ISBN 2 85535 159 6
Code : 467

F 135/86/7